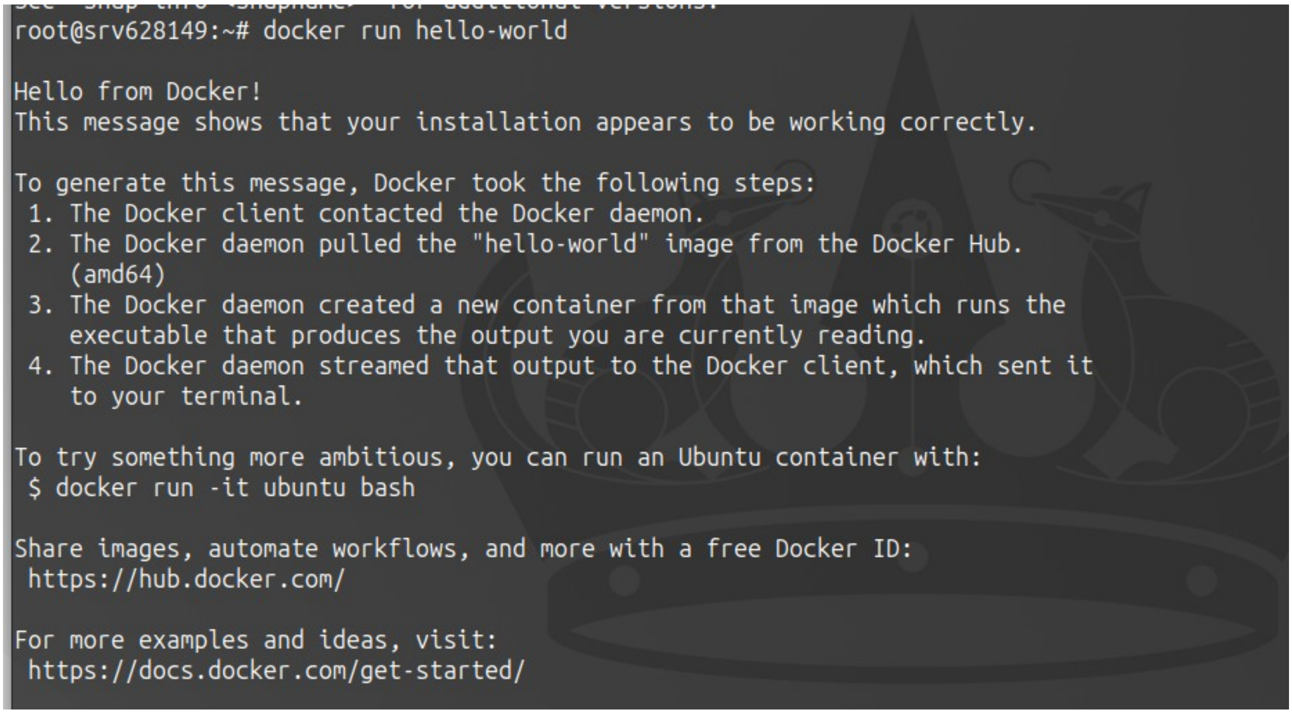
Exercise 1: Getting Started with Containers

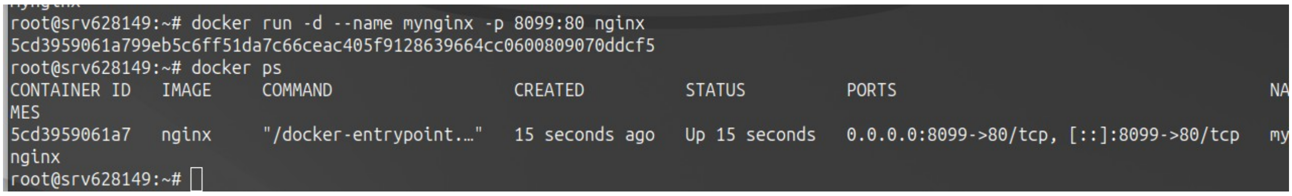
commands used : docker run hello-world
docker run -d --name mynginx -p 8099:80 nginx
docker run -it --name myalpine alpine sh
docker ps
docker ps -a
docker logs
docker exec

docker run hello-world:



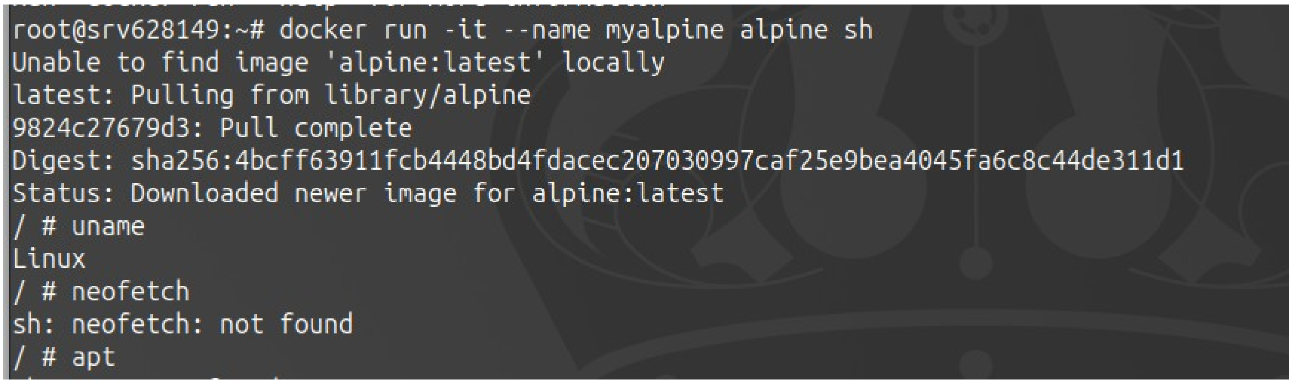
```
root@srv628149:~# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

docker run -d --name mynginx -p 8099:80 nginx:



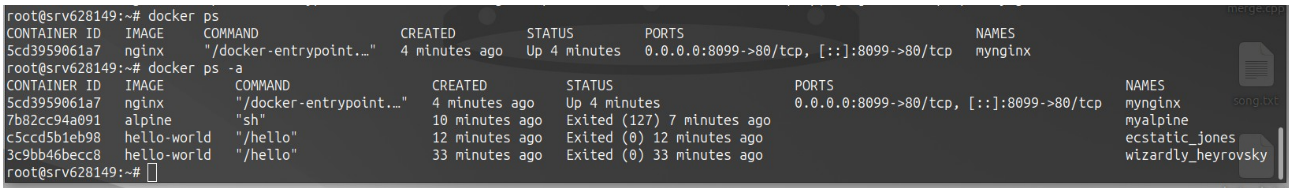
```
root@srv628149:~# docker run -d --name mynginx -p 8099:80 nginx
5cd3959061a799eb5c6ff51da7c66ceac405f9128639664cc0600809070ddcf5
root@srv628149:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                   NA
MES
5cd3959061a7   nginx     "/docker-entrypoint.…"   15 seconds ago   Up 15 seconds   0.0.0.0:8099->80/tcp, [::]:8099->80/tcp   my
nginx
root@srv628149:~# 
```

docker run -it --name myalpine alpine sh:



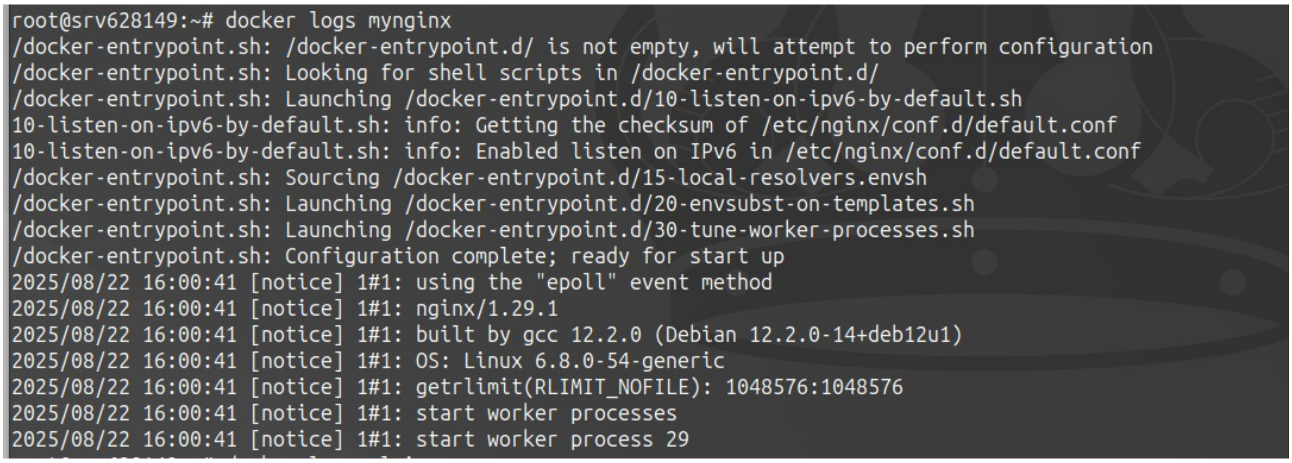
```
root@srv628149:~# docker run -it --name myalpine alpine sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
9824c27679d3: Pull complete
Digest: sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
Status: Downloaded newer image for alpine:latest
/ # uname
Linux
/ # neofetch
sh: neofetch: not found
/ # apt
```

docker ps & docker ps -a:



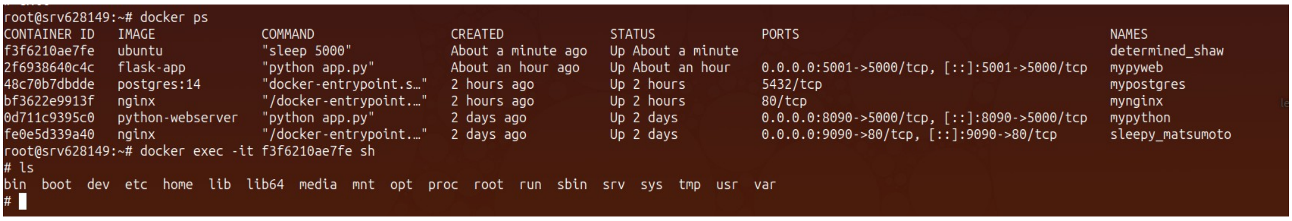
```
root@srv628149:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                     NAMES
5cd3959061a7   nginx     "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes   0.0.0.0:8099->80/tcp, [::]:8099->80/tcp   mynginx
root@srv628149:~# docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS                       PORTS                                     NAMES
5cd3959061a7   nginx         "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes                 0.0.0.0:8099->80/tcp, [::]:8099->80/tcp   mynginx
7b82cc94a091   alpine        "sh"                     10 minutes ago   Exited (127) 7 minutes ago                                             myalpine
c5ccd5b1eb98   hello-world   "/hello"                 12 minutes ago   Exited (0) 12 minutes ago                                              ecstatic_jones
3c9bb46becc8   hello-world   "/hello"                 33 minutes ago   Exited (0) 33 minutes ago                                              wizardly_heyrovsky
root@srv628149:~# 
```

docker logs:



```
root@srv628149:~# docker logs mynginx
/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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
2025/08/22 16:00:41 [notice] 1#1: using the "epoll" event method
2025/08/22 16:00:41 [notice] 1#1: nginx/1.29.1
2025/08/22 16:00:41 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/08/22 16:00:41 [notice] 1#1: OS: Linux 6.8.0-54-generic
2025/08/22 16:00:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/08/22 16:00:41 [notice] 1#1: start worker processes
2025/08/22 16:00:41 [notice] 1#1: start worker process 29
```

docker exec:



```
root@srv628149:~# docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS              PORTS                                         NAMES
f3f6210ae7fe   ubuntu             "sleep 5000"             About a minute ago   Up About a minute                                                 determined_shaw
2f6938640c4c   flask-app          "python app.py"          About an hour ago    Up About an hour    0.0.0.0:5001->5000/tcp, [::]:5001->5000/tcp   mypyweb
48c70b7dbdde   postgres:14        "docker-entrypoint.s…"   2 hours ago          Up 2 hours          5432/tcp                                      mypostgres
bf3622e9913f   nginx              "/docker-entrypoint.…"   2 hours ago          Up 2 hours          80/tcp                                        mynginx
0d711c9395c0   python-webserver   "python app.py"          2 days ago           Up 2 days           0.0.0.0:8090->5000/tcp, [::]:8090->5000/tcp   mypython
fe0e5d339a40   nginx              "/docker-entrypoint.…"   2 days ago           Up 2 days           0.0.0.0:9090->80/tcp, [::]:9090->80/tcp       sleepy_matsumoto
root@srv628149:~# docker exec -it f3f6210ae7fe sh
# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
# 
```

Q1. What happens if a container doesn't run in detached mode?

The container runs in the foreground, and you see its logs directly in your terminal.

Q2. What if ports aren't mapped?

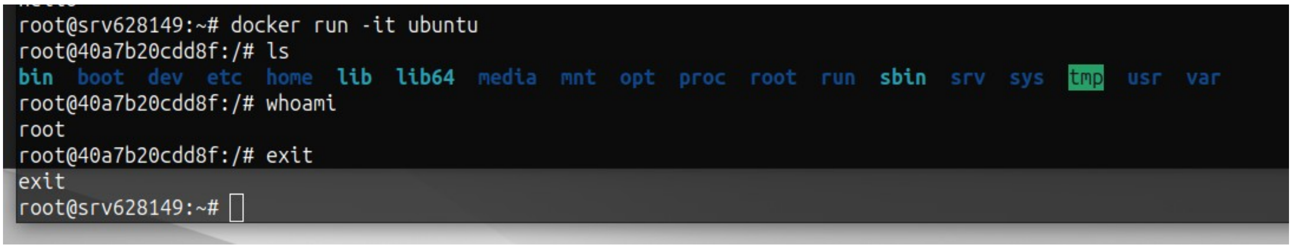
The container still runs, but its service is only accessible inside the container's network.

Exercise 2: Working with Container State

commands used:

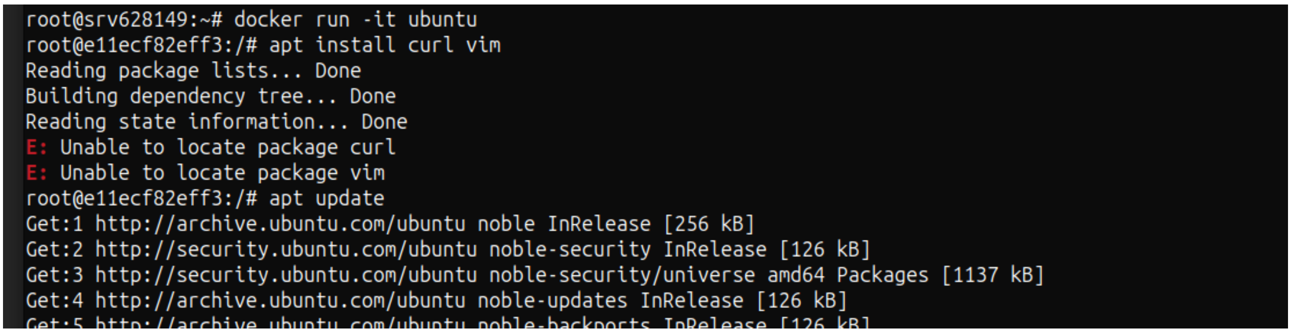
docker run it ubuntu
apt update & apt install curl vim
docker commit <id> ubuntu-tools
docker run -it ubuntu-tools
docker images
docker tag ubuntu-tools ubuntu-tools:v1

docker run -it ubuntu:



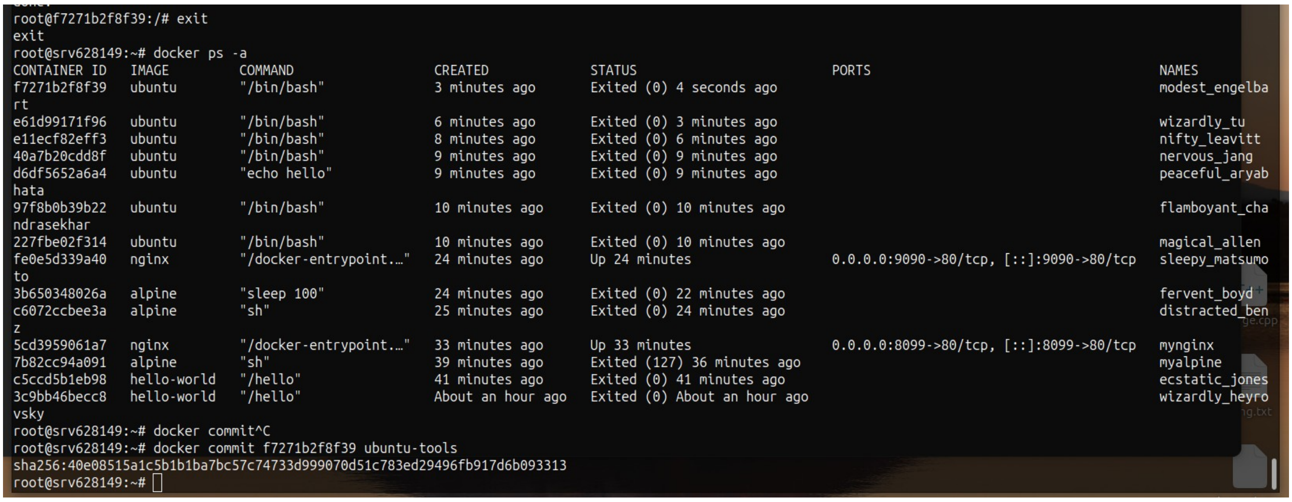
```
root@srv628149:~# docker run -it ubuntu
root@40a7b20cdd8f:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@40a7b20cdd8f:/# whoami
root
root@40a7b20cdd8f:/# exit
exit
root@srv628149:~# 
```

apt update & apt install curl vim:



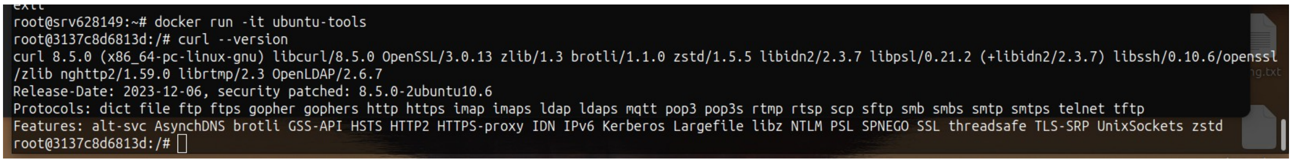
```
root@srv628149:~# docker run -it ubuntu
root@e11ecf82eff3:/# apt install curl vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package curl
E: Unable to locate package vim
root@e11ecf82eff3:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1137 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
```

docker commit <id> ubuntu-tools:



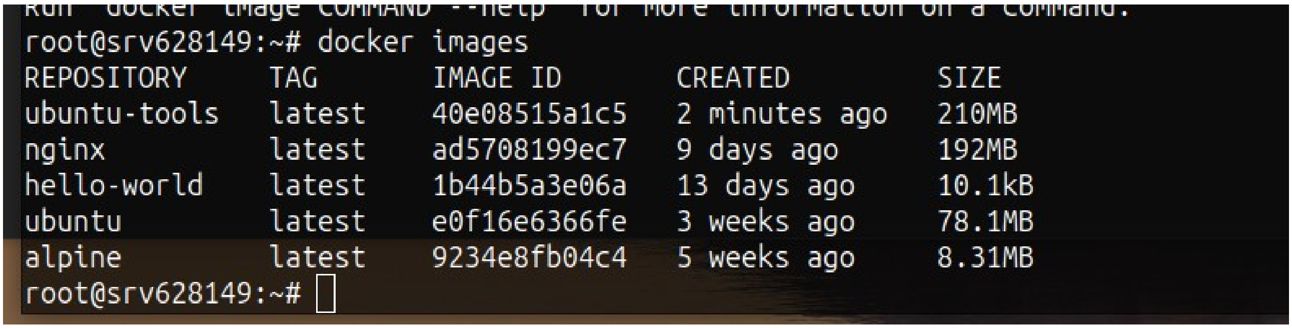
```
root@f7271b2f8f39:/# exit
exit
root@srv628149:~# docker ps -a
CONTAINER ID   IMAGE         COMMAND                   CREATED              STATUS                         PORTS                                         NAMES
f7271b2f8f39   ubuntu        "/bin/bash"               3 minutes ago        Exited (0) 4 seconds ago                                                     modest_engelba
rt
e61d99171f96   ubuntu        "/bin/bash"               6 minutes ago        Exited (0) 3 minutes ago                                                     wizardly_tu
e11ecf82eff3   ubuntu        "/bin/bash"               8 minutes ago        Exited (0) 6 minutes ago                                                     nifty_leavitt
40a7b20cdd8f   ubuntu        "/bin/bash"               9 minutes ago        Exited (0) 9 minutes ago                                                     nervous_jang
d6df5652a6a4   ubuntu        "echo hello"              9 minutes ago        Exited (0) 9 minutes ago                                                     peaceful_aryab
hata
97f8b0b39b22   ubuntu        "/bin/bash"               10 minutes ago       Exited (0) 10 minutes ago                                                    flamboyant_cha
ndrasekhar
227fbe02f314   ubuntu        "/bin/bash"               10 minutes ago       Exited (0) 10 minutes ago                                                    magical_allen
fe0e5d339a40   nginx         "/docker-entrypoint.…"    24 minutes ago       Up 24 minutes                  0.0.0.0:9090->80/tcp, [::]:9090->80/tcp       sleepy_matsumo
to
3b650348026a   alpine        "sleep 100"               24 minutes ago       Exited (0) 22 minutes ago                                                    fervent_boyd
c6072ccbee3a   alpine        "sh"                      25 minutes ago       Exited (0) 24 minutes ago                                                    distracted_ben
z
5cd3959061a7   nginx         "/docker-entrypoint.…"    33 minutes ago       Up 33 minutes                  0.0.0.0:8099->80/tcp, [::]:8099->80/tcp       mynginx
7b82cc94a091   alpine        "sh"                      39 minutes ago       Exited (127) 36 minutes ago                                                  myalpine
c5ccd5b1eb98   hello-world   "/hello"                  41 minutes ago       Exited (0) 41 minutes ago                                                    ecstatic_jones
3c9bb46becc8   hello-world   "/hello"                  About an hour ago    Exited (0) About an hour ago                                                 wizardly_heyro
vsky
root@srv628149:~# docker commit^C
root@srv628149:~# docker commit f7271b2f8f39 ubuntu-tools
sha256:40e08515a1c5b1b1ba7bc57c74733d999070d51c783ed29496fb917d6b093313
root@srv628149:~# 
```

docker run -it ubuntu-tools:



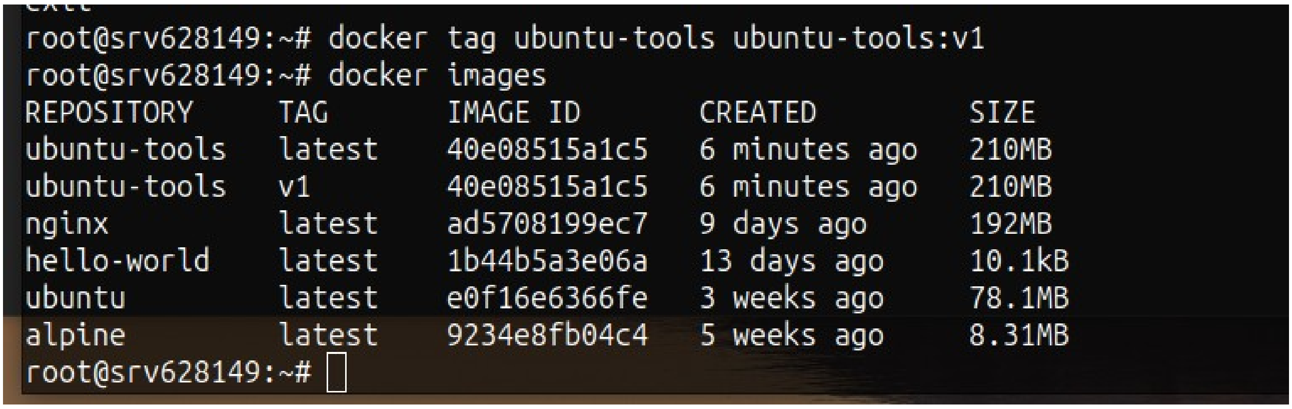
```
root@srv628149:~# docker run -it ubuntu-tools
root@3137c8d6813d:/# curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli/1.1.0 zstd/1.5.5 libidn2/2.3.7 libpsl/0.21.2 (+libidn2/2.3.7) libssh/0.10.6/openssl
/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM PSL SPNEGO SSL threadsafe TLS-SRP UnixSockets zstd
root@3137c8d6813d:/# 
```

docker images:



```
Run 'docker image COMMAND --help' for more information on a command.
root@srv628149:~# docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
ubuntu-tools    latest    40e08515a1c5   2 minutes ago    210MB
nginx           latest    ad5708199ec7   9 days ago       192MB
hello-world     latest    1b44b5a3e06a   13 days ago      10.1kB
ubuntu          latest    e0f16e6366fe   3 weeks ago      78.1MB
alpine          latest    9234e8fb04c4   5 weeks ago      8.31MB
root@srv628149:~# 
```

docker tag ubuntu-tools ubuntu-tools:v1:

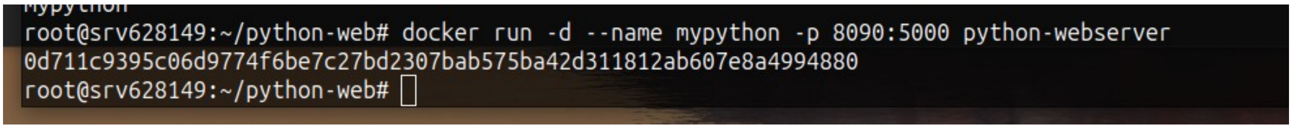


```
exit
root@srv628149:~# docker tag ubuntu-tools ubuntu-tools:v1
root@srv628149:~# docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
ubuntu-tools    latest    40e08515a1c5   6 minutes ago    210MB
ubuntu-tools    v1        40e08515a1c5   6 minutes ago    210MB
nginx           latest    ad5708199ec7   9 days ago       192MB
hello-world     latest    1b44b5a3e06a   13 days ago      10.1kB
ubuntu          latest    e0f16e6366fe   3 weeks ago      78.1MB
alpine          latest    9234e8fb04c4   5 weeks ago      8.31MB
root@srv628149:~# 
```

Exercise 3: Build Custom Images Using Dockerfile

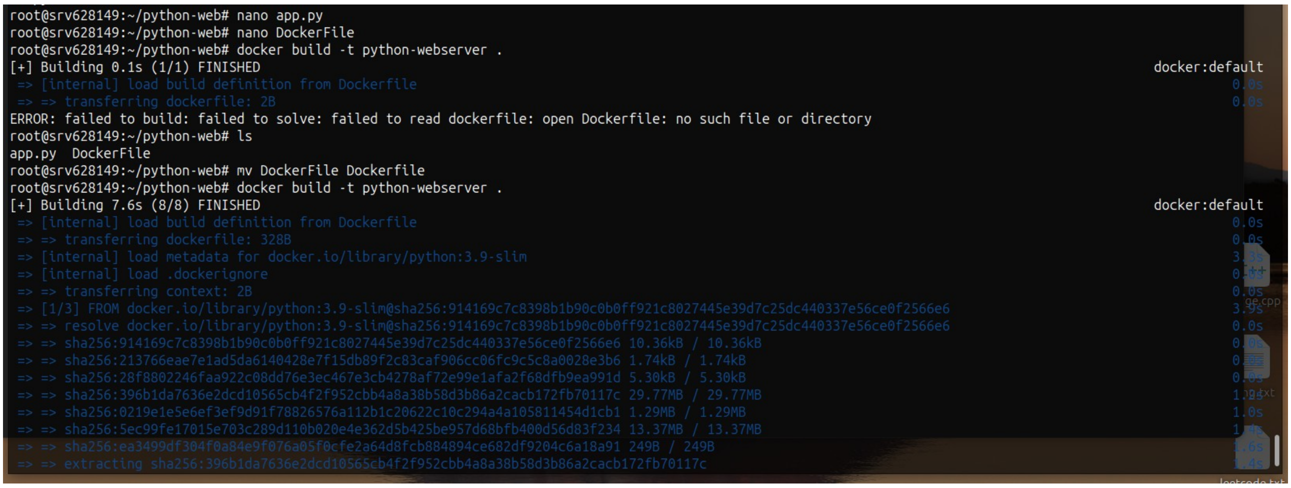
commands used: docker run -d –name mypython -p 8090:5000 python-webserver
docker build -t python-webserver
curl http://localhost:8090/

docker run -d –name mypython -p 8090:5000 python-webserver:



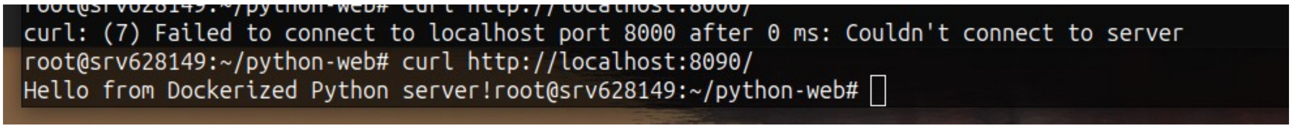
```
mypython
root@srv628149:~/python-web# docker run -d --name mypython -p 8090:5000 python-webserver
0d711c9395c06d9774f6be7c27bd2307bab575ba42d311812ab607e8a4994880
root@srv628149:~/python-web# 
```

docker build -t python-webserver:



```
root@srv628149:~/python-web# nano app.py
root@srv628149:~/python-web# nano DockerFile
root@srv628149:~/python-web# docker build -t python-webserver .
[+] Building 0.1s (1/1) FINISHED                                                                                        docker:default
 => [internal] load build definition from Dockerfile                                                                              0.0s
 => => transferring dockerfile: 2B                                                                                                0.0s
ERROR: failed to build: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
root@srv628149:~/python-web# ls
app.py  DockerFile
root@srv628149:~/python-web# mv DockerFile Dockerfile
root@srv628149:~/python-web# docker build -t python-webserver .
[+] Building 7.6s (8/8) FINISHED                                                                                        docker:default
 => [internal] load build definition from Dockerfile                                                                              0.0s
 => => transferring dockerfile: 328B                                                                                              0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                                                3.3s
 => [internal] load .dockerignore                                                                                                 0.0s
 => => transferring context: 2B                                                                                                   0.0s
 => [1/3] FROM docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0f2566e6          3.9s
 => => resolve docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0f2566e6          0.0s
 => => sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0f2566e6 10.36kB / 10.36kB                                  0.0s
 => => sha256:213766eae7e1ad5da6140428e7f15db89f2c83caf906cc06fc9c5c8a0028e3b6 1.74kB / 1.74kB                                    0.0s
 => => sha256:28f8802246faa922c08dd76e3ec467e3cb4278af72e99e1afa2f68dfb9ea991d 5.30kB / 5.30kB                                    0.0s
 => => sha256:396b1da7636e2dcd10565cb4f2f952cbb4a8a38b58d3b86a2cacb172fb70117c 29.77MB / 29.77MB                                  1.2s
 => => sha256:0219e1e5e6ef3ef9d91f78826576a112b1c20622c10c294a4a105811454d1cb1 1.29MB / 1.29MB                                    1.0s
 => => sha256:5ec99fe17015e703c289d110b020e4e362d5b425be957d68bfb400d56d83f234 13.37MB / 13.37MB                                  1.4s
 => => sha256:ea3499df304f0a84e9f076a05f0cfe2a64d8fcb884894ce682df9204c6a18a91 249B / 249B                                        1.6s
 => => extracting sha256:396b1da7636e2dcd10565cb4f2f952cbb4a8a38b58d3b86a2cacb172fb70117c                                         1.4s
```

curl http://localhost:8090/ :

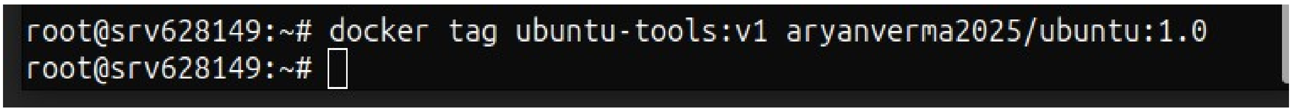


```
root@srv628149:~/python-web# curl http://localhost:8000/
curl: (7) Failed to connect to localhost port 8000 after 0 ms: Couldn't connect to server
root@srv628149:~/python-web# curl http://localhost:8090/
Hello from Dockerized Python server!root@srv628149:~/python-web# 
```

Exercise 4: Sharing Images

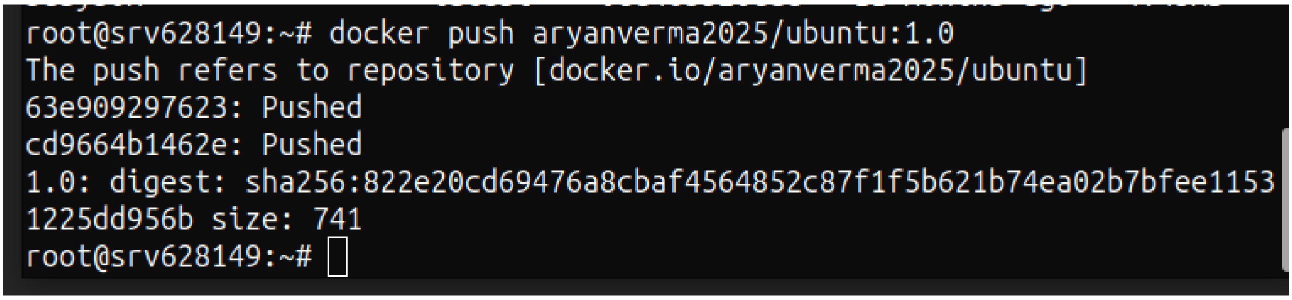
commands used:
docker tag ubuntu-tools:v1 aryanverma2025/ubuntu:1.0
docker push aryanverma2025/ubuntu:1.0
docker pull aryanverma2025/ubuntu:1.0

docker tag ubuntu-tools:v1 aryanverma2025/ubuntu:1.0:



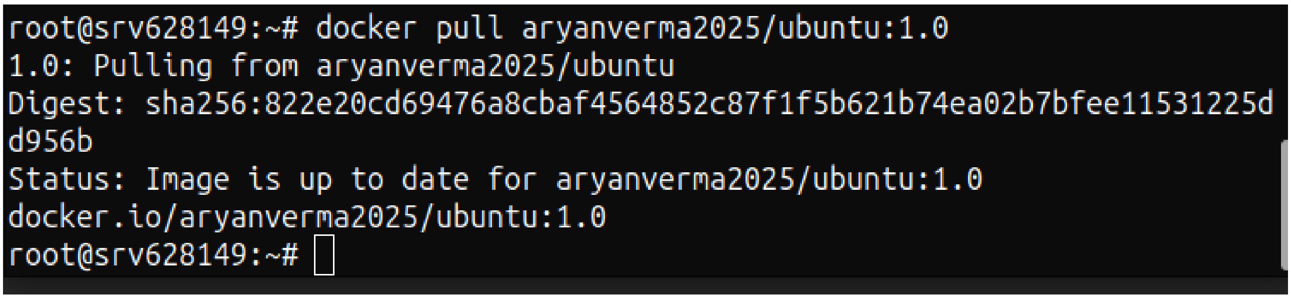
```
root@srv628149:~# docker tag ubuntu-tools:v1 aryanverma2025/ubuntu:1.0
root@srv628149:~# 
```

docker push aryanverma2025/ubuntu:1.0:



```
root@srv628149:~# docker push aryanverma2025/ubuntu:1.0
The push refers to repository [docker.io/aryanverma2025/ubuntu]
63e909297623: Pushed
cd9664b1462e: Pushed
1.0: digest: sha256:822e20cd69476a8cbaf4564852c87f1f5b621b74ea02b7bfee1153
1225dd956b size: 741
root@srv628149:~# 
```

docker pull aryanverma2025/ubuntu:1.0:



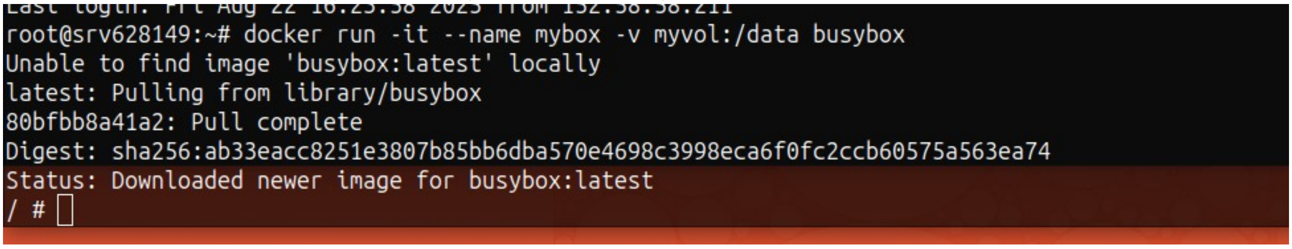
```
root@srv628149:~# docker pull aryanverma2025/ubuntu:1.0
1.0: Pulling from aryanverma2025/ubuntu
Digest: sha256:822e20cd69476a8cbaf4564852c87f1f5b621b74ea02b7bfee11531225d
d956b
Status: Image is up to date for aryanverma2025/ubuntu:1.0
docker.io/aryanverma2025/ubuntu:1.0
root@srv628149:~# 
```

Exercise 5: Data persistance with volumes

commands used:

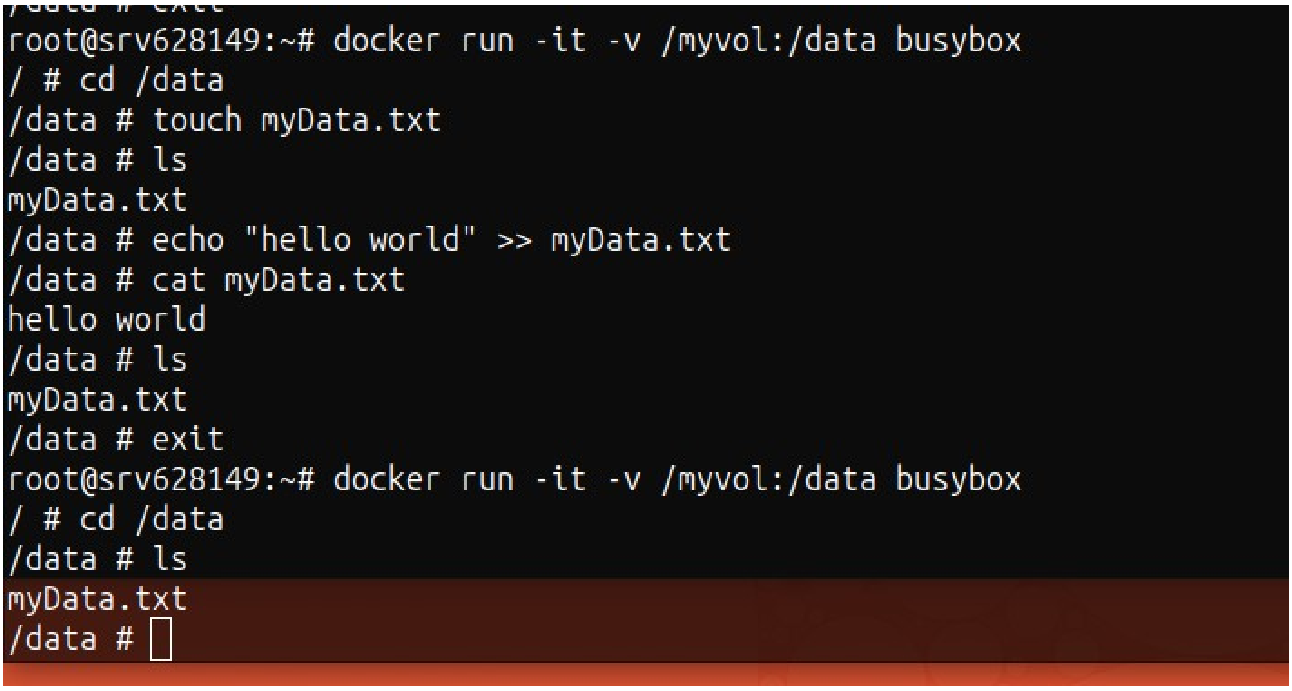
docker run -it –name mybox -v myvol:/data busybox
docker run -it -v myvol:/data busybox
docker run -it -v ${pwd}/data:/data busybox

docker run -it –name mybox -v myvol:/data busybox:



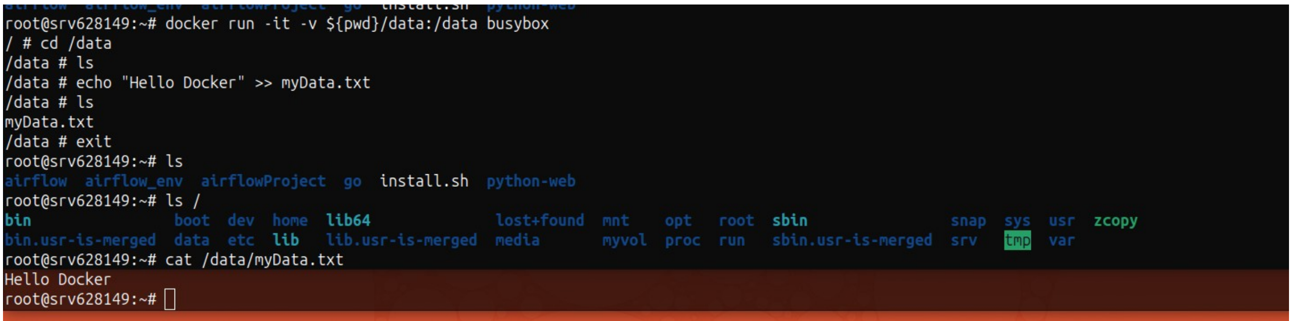
```
root@srv628149:~# docker run -it --name mybox -v myvol:/data busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
80bfbb8a41a2: Pull complete
Digest: sha256:ab33eacc8251e3807b85bb6dba570e4698c3998eca6f0fc2ccb60575a563ea74
Status: Downloaded newer image for busybox:latest
/ # 
```

docker run -it -v myvol:/data busybox:



```
root@srv628149:~# docker run -it -v /myvol:/data busybox
/ # cd /data
/data # touch myData.txt
/data # ls
myData.txt
/data # echo "hello world" >> myData.txt
/data # cat myData.txt
hello world
/data # ls
myData.txt
/data # exit
root@srv628149:~# docker run -it -v /myvol:/data busybox
/ # cd /data
/data # ls
myData.txt
/data # 
```

docker run -it -v ${pwd}/data:/data busybox

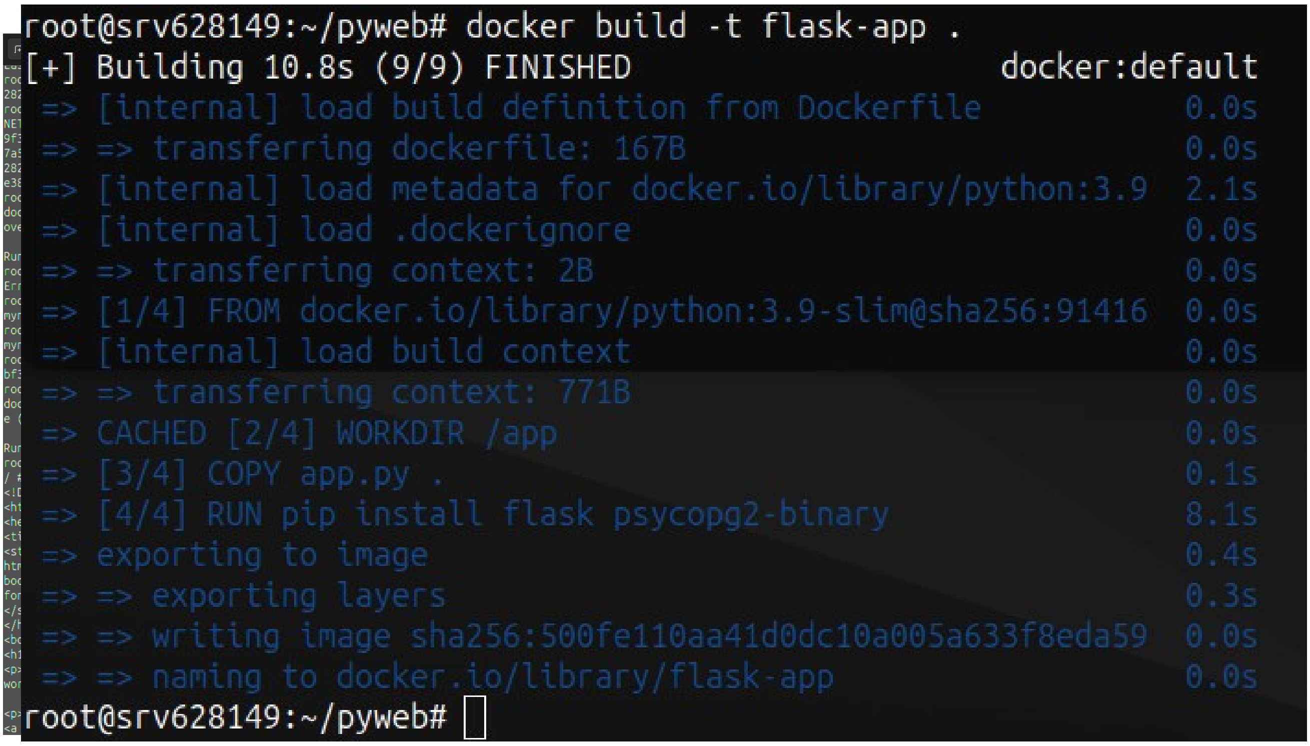


```
root@srv628149:~# docker run -it -v ${pwd}/data:/data busybox
/ # cd /data
/data # ls
/data # echo "Hello Docker" >> myData.txt
/data # ls
myData.txt
/data # exit
root@srv628149:~# ls
airflow  airflow_env  airflowProject  go  install.sh  python-web
root@srv628149:~# ls /
bin                boot  dev  home  lib64              lost+found  mnt    opt   root  sbin                snap  sys  usr  zcopy
bin.usr-is-merged  data  etc  lib   lib.usr-is-merged  media       myvol  proc  run   sbin.usr-is-merged  srv   tmp  var
root@srv628149:~# cat /data/myData.txt
Hello Docker
root@srv628149:~# 
```

Exercise 6: Container Networking Basics

commands used:

docker network create mynetwork:
docker network ls
docker run -d –name mynginx –network mynetwork nginx
docker run -it –name mybox –network mynetwork busybox
wget -q0- http://mynginx

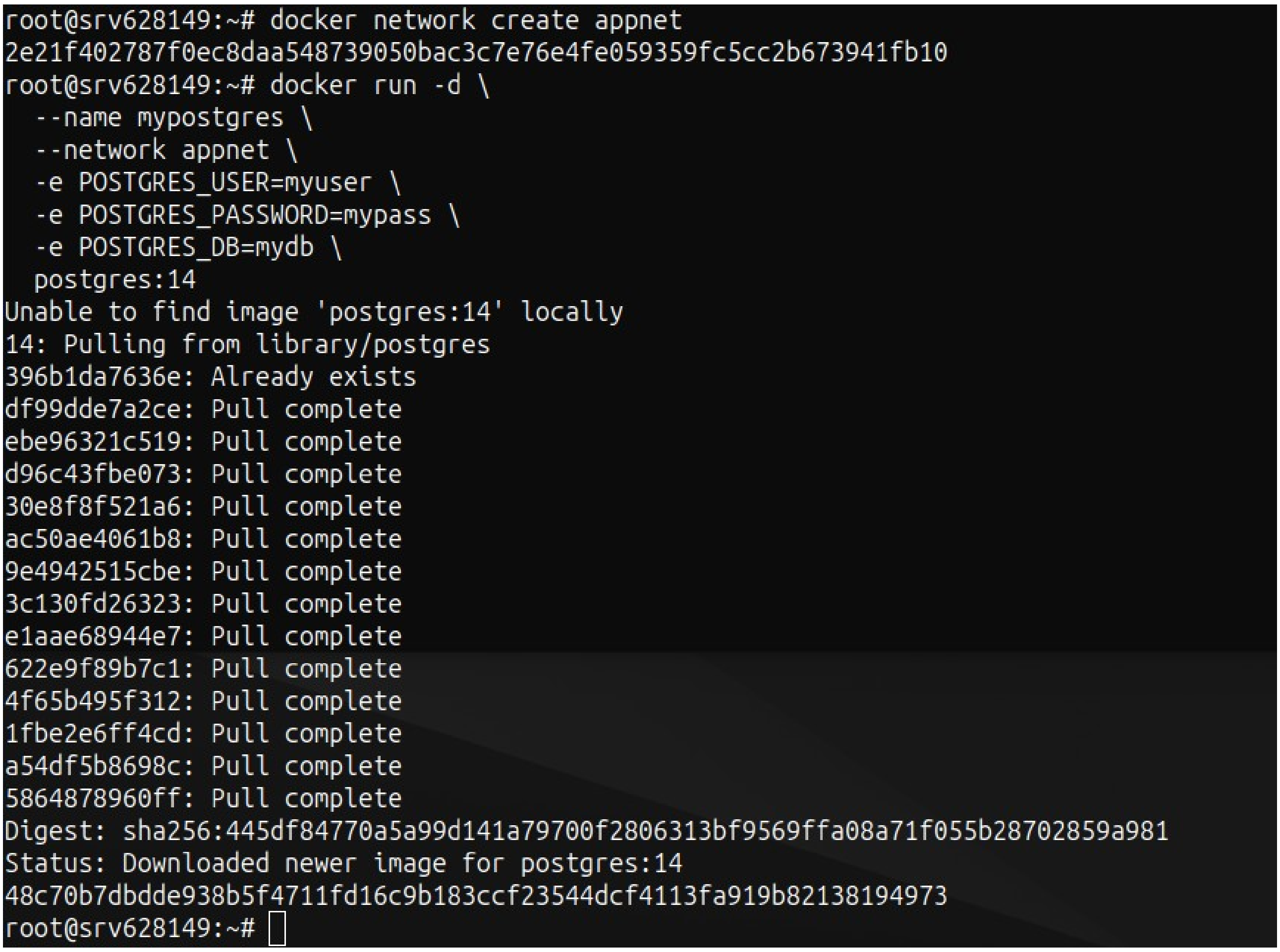


```
root@srv628149:~/pyweb# docker build -t flask-app .
[+] Building 10.8s (9/9) FINISHED                          docker:default
 => [internal] load build definition from Dockerfile                 0.0s
 => => transferring dockerfile: 167B                                 0.0s
 => [internal] load metadata for docker.io/library/python:3.9        2.1s
 => [internal] load .dockerignore                                    0.0s
 => => transferring context: 2B                                      0.0s
 => [1/4] FROM docker.io/library/python:3.9-slim@sha256:91416        0.0s
 => [internal] load build context                                    0.0s
 => => transferring context: 771B                                    0.0s
 => CACHED [2/4] WORKDIR /app                                        0.0s
 => [3/4] COPY app.py .                                              0.1s
 => [4/4] RUN pip install flask psycopg2-binary                      8.1s
 => exporting to image                                               0.4s
 => => exporting layers                                              0.3s
 => => writing image sha256:500fe110aa41d0dc10a005a633f8eda59        0.0s
 => => naming to docker.io/library/flask-app                         0.0s
root@srv628149:~/pyweb# 
```

Exercise 7: Building a two tier app

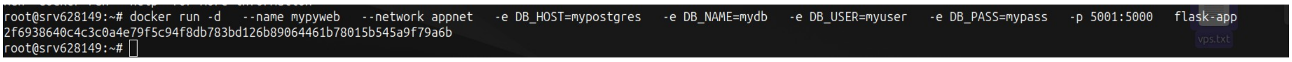
commands used: docker network create appnet
docker run -d –name myppyweb –network appnet -e DB_HOST=mypostgres -e DB_NAME=mydb -e DB_PASS=mypass -p 5001:5000 flask-app
curl http://localhost:5001:

docker network create appnet:



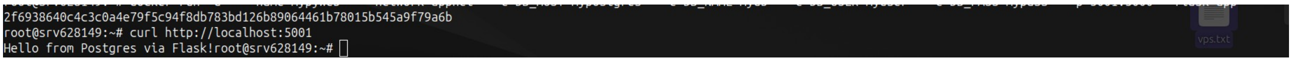
```
root@srv628149:~# docker network create appnet
2e21f402787f0ec8daa548739050bac3c7e76e4fe059359fc5cc2b673941fb10
root@srv628149:~# docker run -d \
  --name mypostgres \
  --network appnet \
  -e POSTGRES_USER=myuser \
  -e POSTGRES_PASSWORD=mypass \
  -e POSTGRES_DB=mydb \
  postgres:14
Unable to find image 'postgres:14' locally
14: Pulling from library/postgres
396b1da7636e: Already exists
df99dde7a2ce: Pull complete
ebe96321c519: Pull complete
d96c43fbe073: Pull complete
30e8f8f521a6: Pull complete
ac50ae4061b8: Pull complete
9e4942515cbe: Pull complete
3c130fd26323: Pull complete
e1aae68944e7: Pull complete
622e9f89b7c1: Pull complete
4f65b495f312: Pull complete
1fbe2e6ff4cd: Pull complete
a54df5b8698c: Pull complete
5864878960ff: Pull complete
Digest: sha256:445df84770a5a99d141a79700f2806313bf9569ffa08a71f055b28702859a981
Status: Downloaded newer image for postgres:14
48c70b7dbdde938b5f4711fd16c9b183ccf23544dcf4113fa919b82138194973
root@srv628149:~# 
```

docker run -d –name myppyweb –network appnet -e DB_HOST=mypostgres -e DB_NAME=mydb -e DB_PASS=mypass -p 5001:5000 flask-app:



```
root@srv628149:~# docker run -d   --name mypyweb   --network appnet   -e DB_HOST=mypostgres   -e DB_NAME=mydb   -e DB_USER=myuser   -e DB_PASS=mypass   -p 5001:5000   flask-app
2f6938640c4c3c0a4e79f5c94f8db783bd126b89064461b78015b545a9f79a6b
root@srv628149:~# 
```

curl http://localhost:5001:

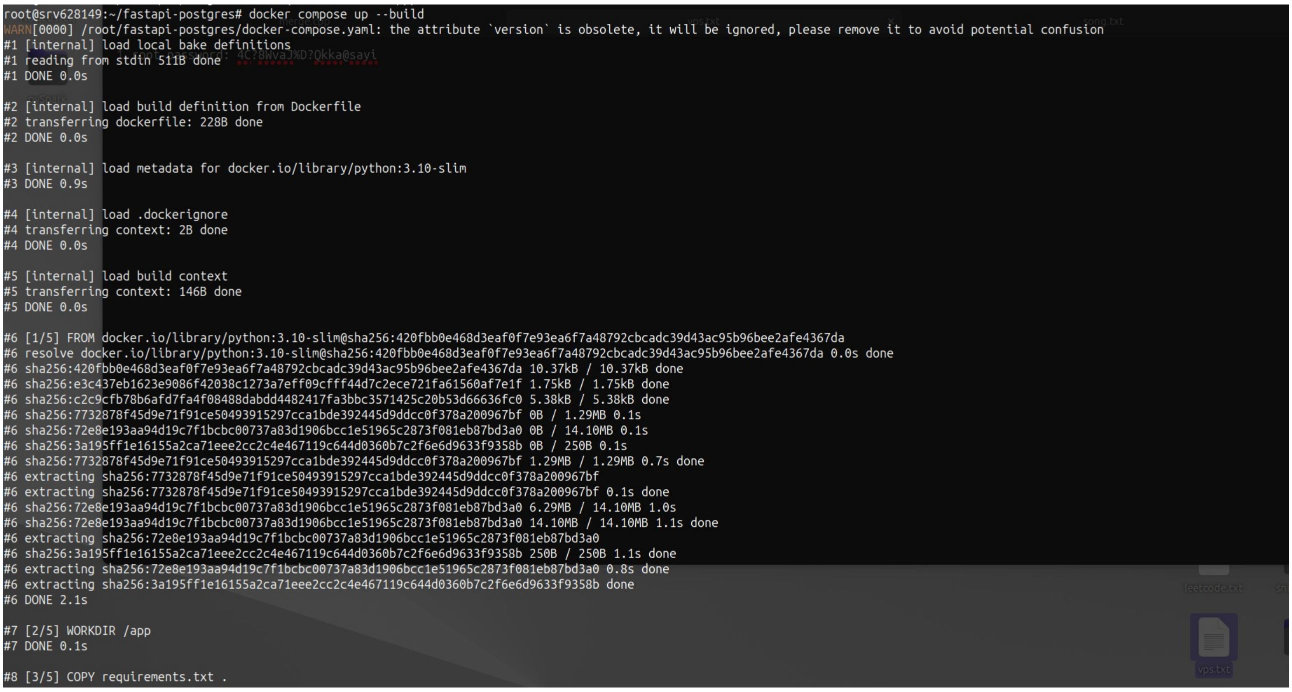


```
2f6938640c4c3c0a4e79f5c94f8db783bd126b89064461b78015b545a9f79a6b
root@srv628149:~# curl http://localhost:5001
Hello from Postgres via Flask!root@srv628149:~# 
```

Exercise 8: Docker Compose Basics

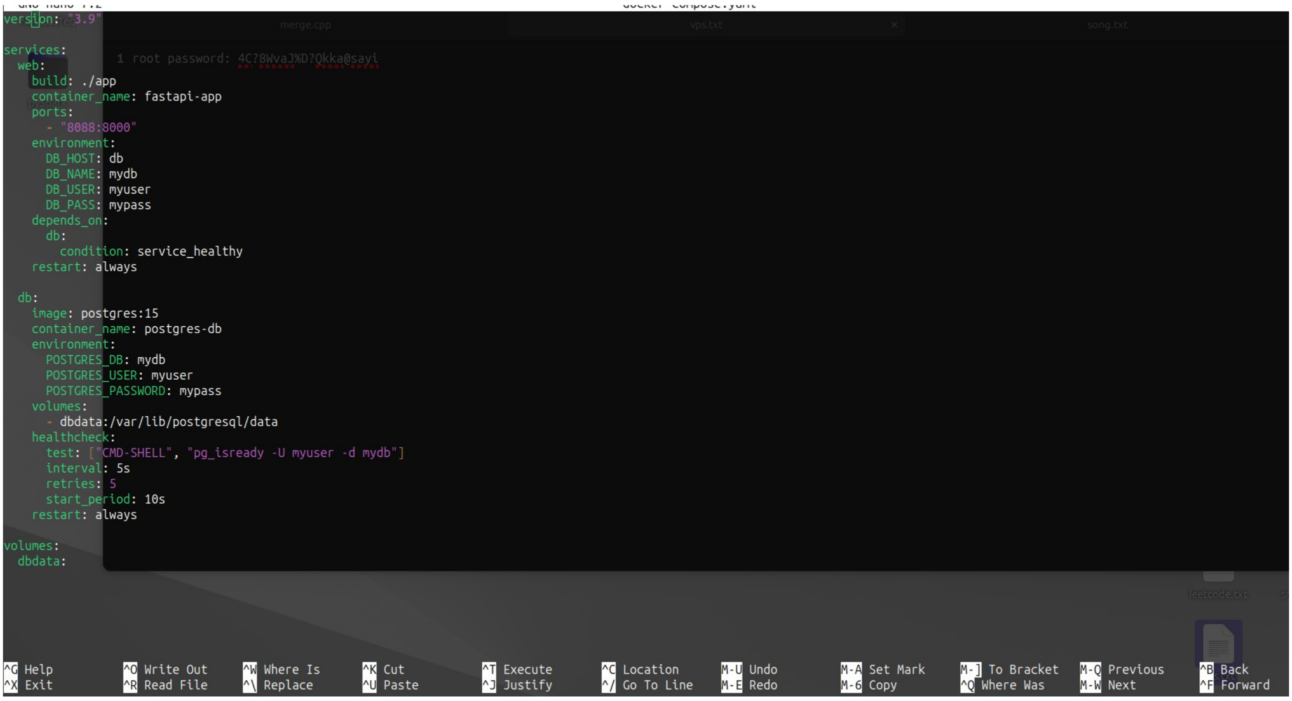
commands used: docker compose up –build
docker compose up
curl http://localhost:8088

docker compose up –build:



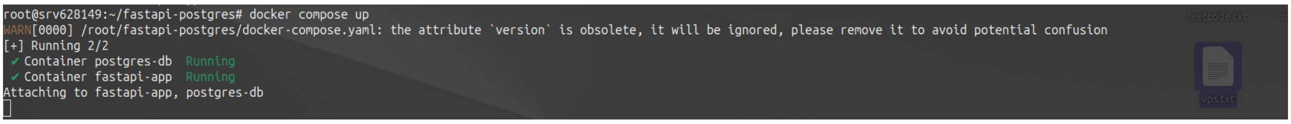
```
root@srv628149:~/fastapi-postgres# docker compose up --build
WARN[0000] /root/fastapi-postgres/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
#1 [internal] load local bake definitions
#1 reading from stdin 511B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 228B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/python:3.10-slim
#3 DONE 0.9s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 146B done
#5 DONE 0.0s

#6 [1/5] FROM docker.io/library/python:3.10-slim@sha256:420fbb0e468d3eaf0f7e93ea6f7a48792cbcadc39d43ac95b96bee2afe4367da
#6 resolve docker.io/library/python:3.10-slim@sha256:420fbb0e468d3eaf0f7e93ea6f7a48792cbcadc39d43ac95b96bee2afe4367da 0.0s done
#6 sha256:420fbb0e468d3eaf0f7e93ea6f7a48792cbcadc39d43ac95b96bee2afe4367da 10.37kB / 10.37kB done
#6 sha256:e3c437eb1623e9086f42038c1273a7eff09cfff44d7c2ece721fa61560af7e1f 1.75kB / 1.75kB done
#6 sha256:c2c9cfb78b6afd7fa4f08488dabdd4482417fa3bbc3571425c20b53d66636fc0 5.38kB / 5.38kB done
#6 sha256:7732878f45d9e71f91ce50493915297cca1bde392445d9ddcc0f378a200967bf 0B / 1.29MB 0.1s
#6 sha256:72e8e193aa94d19c7f1bcbc00737a83d1906bcc1e51965c2873f081eb87bd3a0 0B / 14.10MB 0.1s
#6 sha256:3a195ff1e16155a2ca71eee2cc2c4e467119c644d0360b7c2f6e6d9633f9358b 0B / 250B 0.1s
#6 sha256:7732878f45d9e71f91ce50493915297cca1bde392445d9ddcc0f378a200967bf 1.29MB / 1.29MB 0.7s done
#6 extracting sha256:7732878f45d9e71f91ce50493915297cca1bde392445d9ddcc0f378a200967bf
#6 extracting sha256:7732878f45d9e71f91ce50493915297cca1bde392445d9ddcc0f378a200967bf 0.1s done
#6 sha256:72e8e193aa94d19c7f1bcbc00737a83d1906bcc1e51965c2873f081eb87bd3a0 6.29MB / 14.10MB 1.0s
#6 sha256:72e8e193aa94d19c7f1bcbc00737a83d1906bcc1e51965c2873f081eb87bd3a0 14.10MB / 14.10MB 1.1s done
#6 extracting sha256:72e8e193aa94d19c7f1bcbc00737a83d1906bcc1e51965c2873f081eb87bd3a0
#6 sha256:3a195ff1e16155a2ca71eee2cc2c4e467119c644d0360b7c2f6e6d9633f9358b 250B / 250B 1.1s done
#6 extracting sha256:72e8e193aa94d19c7f1bcbc00737a83d1906bcc1e51965c2873f081eb87bd3a0 0.8s done
#6 extracting sha256:3a195ff1e16155a2ca71eee2cc2c4e467119c644d0360b7c2f6e6d9633f9358b done
#6 DONE 2.1s

#7 [2/5] WORKDIR /app
#7 DONE 0.1s

#8 [3/5] COPY requirements.txt .
```

.yaml file:



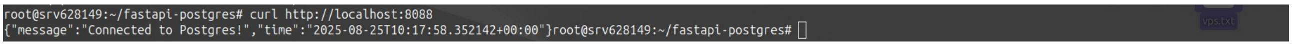
```
version: "3.9"

services:
  web:
    build: ./app
    container_name: fastapi-app
    ports:
      - "8088:8000"
    environment:
      DB_HOST: db
      DB_NAME: mydb
      DB_USER: myuser
      DB_PASS: mypass
    depends_on:
      db:
        condition: service_healthy
    restart: always

  db:
    image: postgres:15
    container_name: postgres-db
    environment:
      POSTGRES_DB: mydb
      POSTGRES_USER: myuser
      POSTGRES_PASSWORD: mypass
    volumes:
      - dbdata:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U myuser -d mydb"]
      interval: 5s
      retries: 5
      start_period: 10s
    restart: always

volumes:
  dbdata:
```

docker compose up:



```
root@srv628149:~/fastapi-postgres# docker compose up
WARN[0000] /root/fastapi-postgres/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 2/2
 ✔ Container postgres-db  Running
 ✔ Container fastapi-app  Running
Attaching to fastapi-app, postgres-db
```

curl http://localhost:8088:

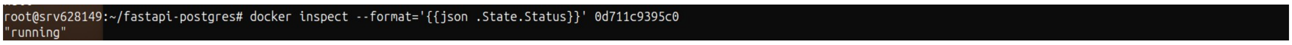


```
root@srv628149:~/fastapi-postgres# curl http://localhost:8088
{"message":"Connected to Postgres!","time":"2025-08-25T10:17:58.352142+00:00"}root@srv628149:~/fastapi-postgres#
```

Exercise 9: Healthchecks and best practices

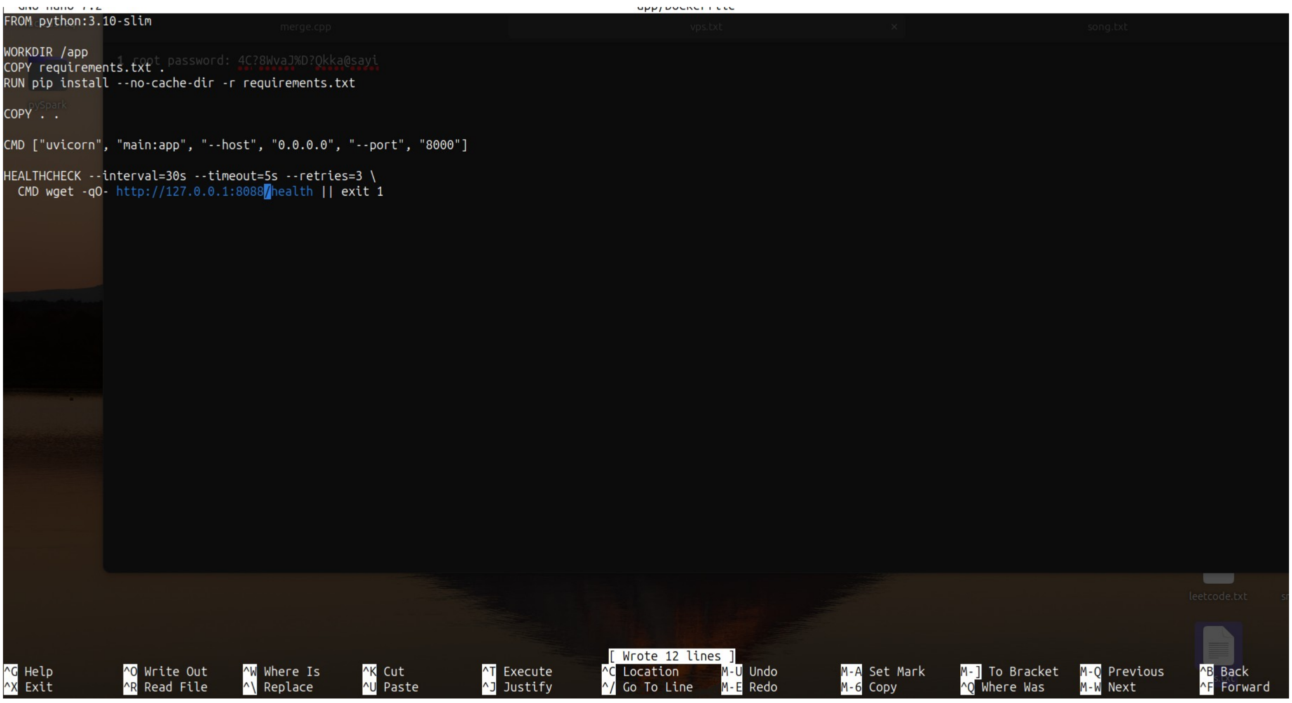
commands used: docker inspect –format'{json .State.Status}}' 0d711c9395c0

docker inspect –format'{json .State.Status}}' 0d711c9395c0:



```
root@srv628149:~/fastapi-postgres# docker inspect --format='{{json .State.Status}}' 0d711c9395c0
"running"
```

Dockerfile with HEALTHCHECK



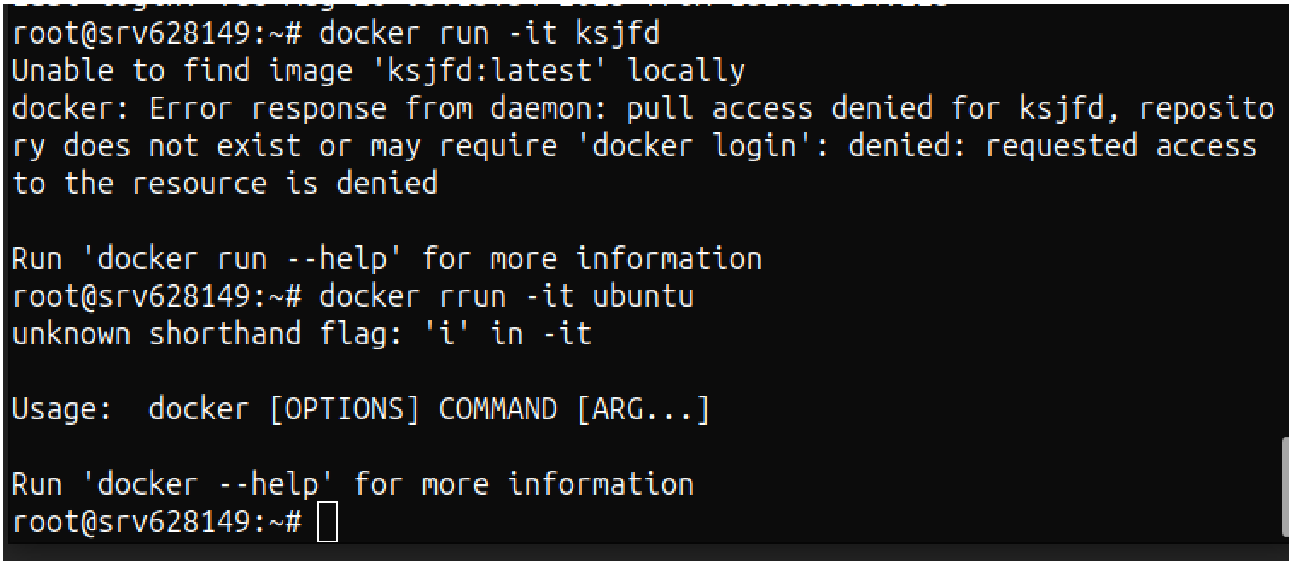
```
FROM python:3.10-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]

HEALTHCHECK --interval=30s --timeout=5s --retries=3 \
  CMD wget -qO- http://127.0.0.1:8088/health || exit 1
```

[Wrote 12 lines]

Exercise 10: Debuggin, Cleanup and Troubleshooting

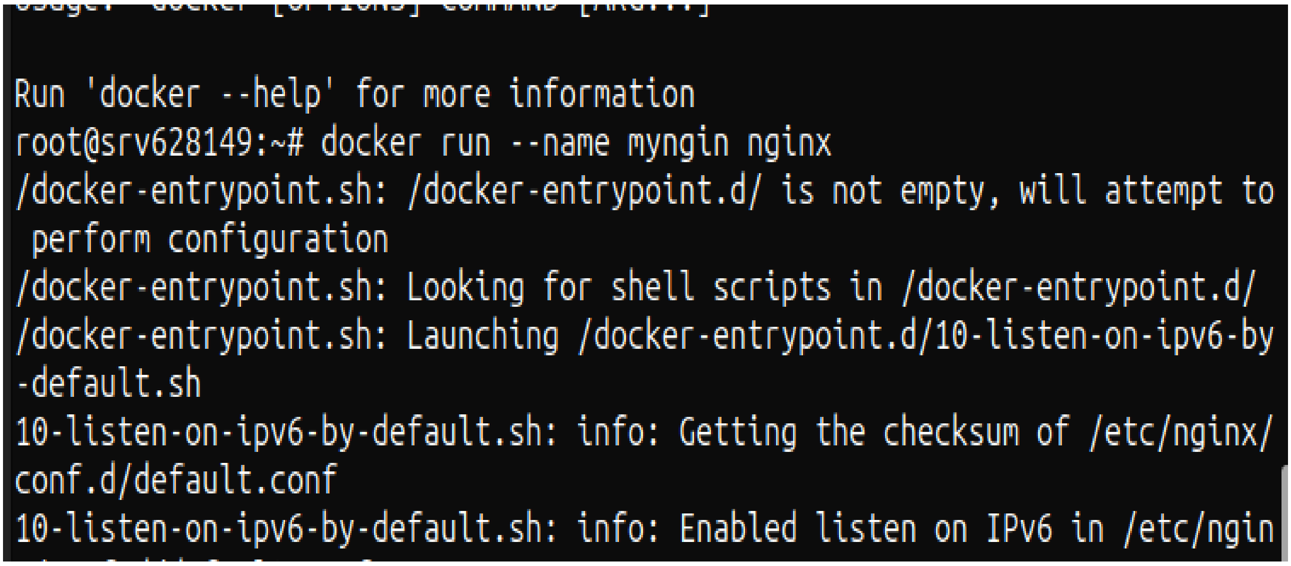
commands used: docker run -it ksjfd
docker run –name mygin nginx

docker run -it ksjfd:



```
root@srv628149:~# docker run -it ksjfd
Unable to find image 'ksjfd:latest' locally
docker: Error response from daemon: pull access denied for ksjfd, reposito
ry does not exist or may require 'docker login': denied: requested access
to the resource is denied

Run 'docker run --help' for more information
root@srv628149:~# docker rrun -it ubuntu
unknown shorthand flag: 'i' in -it

Usage:  docker [OPTIONS] COMMAND [ARG...]

Run 'docker --help' for more information
root@srv628149:~# 
```

docker run –name mygin nginx:
(container runs but cannot access it using outside network)

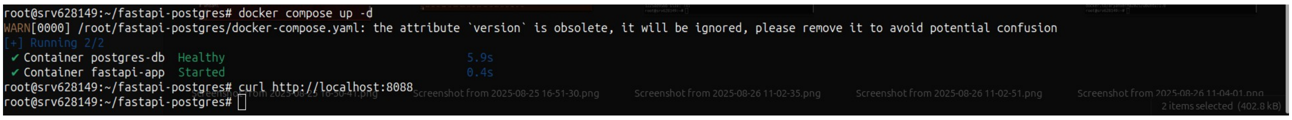


```
Usage:  docker [OPTIONS] COMMAND [ARG...]

Run 'docker --help' for more information
root@srv628149:~# docker run --name myngin nginx
/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/ngin
```

Capstone Project: Dockerizing FastAPI + Postgres

commands used: docker compose up –build
curl http://localhost:8088/

docker compose up –build &
curl http://localhost:8088/:



```
root@srv628149:~/fastapi-postgres# docker compose up -d
WARN[0000] /root/fastapi-postgres/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 2/2
 ✔ Container postgres-db  Healthy                                    5.9s
 ✔ Container fastapi-app  Started                                    0.4s
root@srv628149:~/fastapi-postgres# curl http://localhost:8088
root@srv628149:~/fastapi-postgres# 
```

Dockerfile:

```
FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8088"]
```

docker-compose.yaml:

```
version: "3.9"

services:
  web:
    build: ./app
    container_name: fastapi_app
    ports:
      - "8000:8000"
    depends_on:
      - db
    environment:
      - DATABASE_HOST=db
      - DATABASE_USER=myuser
      - DATABASE_PASSWORD=mypassword
      - DATABASE_NAME=mydb
    networks:
      - app-network

  db:
    image: postgres:14
    container_name: postgres_db
    restart: always
    environment:
      POSTGRES_USER: myuser
      POSTGRES_PASSWORD: mypassword
      POSTGRES_DB: mydb
    volumes:
      - db_data:/var/lib/postgresql/data
    networks:
      - app-network

volumes:
  db_data:

networks:
  app-network:
    driver: bridge
```