JOSH

JOSH.KIDFILEAPP@GMAIL.COM

302 235 9992

DELAWARE

DOCKER COMPOSE

Docker Compose

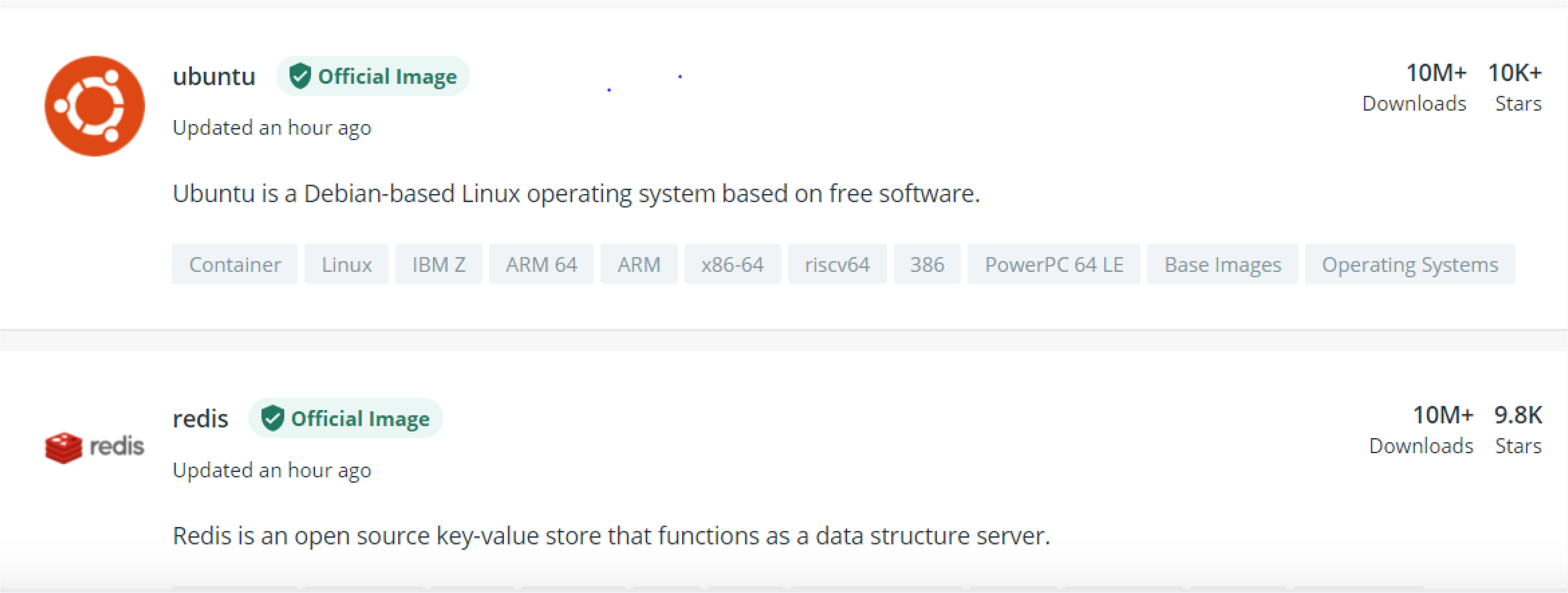
- Tool for **defining** & **running multi-container docker** applications.

- Use **yaml files** to configure application services **(docker-compose.yml).**

- Can start **all services** with **a single command** : **docker compose up**

- Can **stop** all services with **a single command** : **docker compose down**

- Can scale up selected services when required: Up to as many containers as possible



ubuntu Official Image

Updated an hour ago

10M+ Downloads 10K+ Stars

Ubuntu is a Debian-based Linux operating system based on free software.

redis Official Image

Updated an hour ago

10M+ Downloads 9.8K Stars

Redis is an open source key-value store that functions as a data structure server.

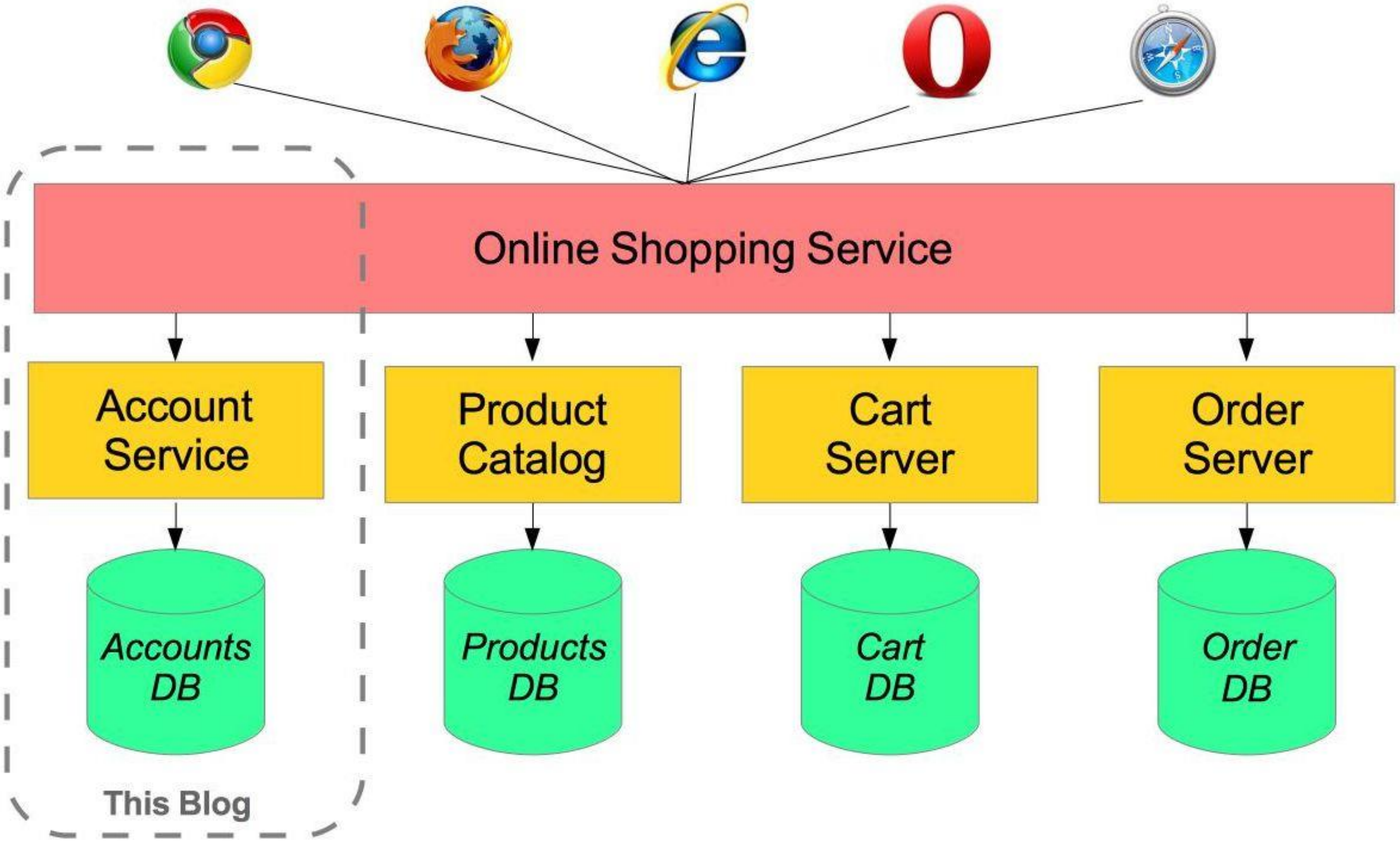
- ❖ In the previous PowerPoint, we learned how to create containers from the images.

- ❖ In the enterprise environment, running one image is not enough.

- ❖ A complete application must be created in which different images must interact to fulfill different services.

- ❖ For instance, we can create a complete application that will deliver different services to the enterprise using **Redis** and **Ubuntu** image/containers.

This Enterprise Environment can Be Created Using Docker Compose for Effortless Scaling



Let take the example of Amazon.com.

Looking at the image, we can see different modules i.e

- Account Service,
- product Service,
- Cart Server

The Account Service has the following:

- Complete Application with different servers
- database, and
- other applications.

Each module is separate, which makes it efficient to make changes.

Also, scaling is effortless

- For instance, in thanksgiving week, more and more people are shopping online, and we need to scale the Product Service Module.
- We can do this easily [with a single command] and without touching any other module, making docker-compose vital.

Let See How This Enterprise Environment is Created Using Docker Compose

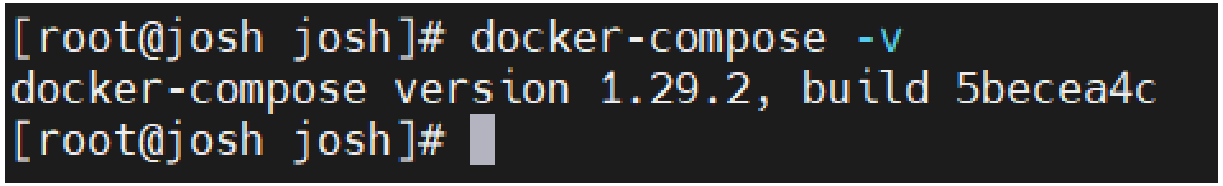
Step 1 : install docker compose
(already installed on windows and mac with docker)

Since you downloaded docker, you should have docker-compose. Come as part of docker.

Check the version of your docker-compose by running the command.

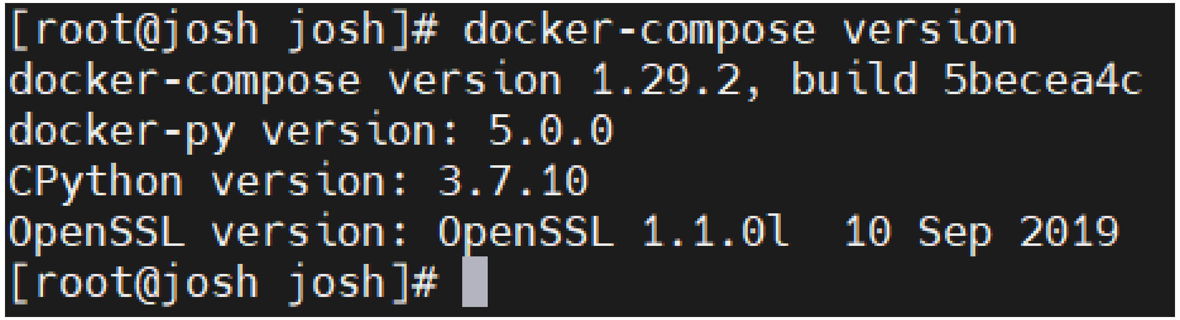
# docker-compose –v && # docker-compose version

# docker-compose –v



```
[root@josh josh]# docker-compose -v
docker-compose version 1.29.2, build 5becea4c
[root@josh josh]#
```

# docker-compose version



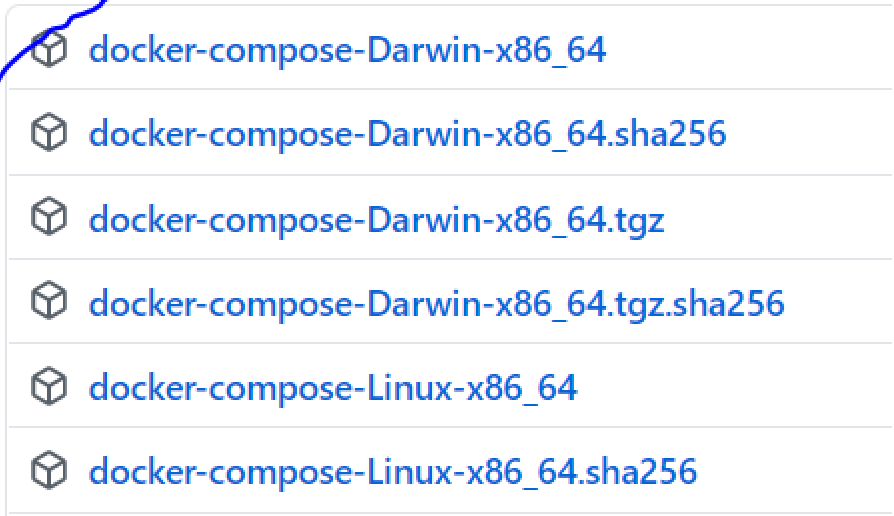
```
[root@josh josh]# docker-compose version
docker-compose version 1.29.2, build 5becea4c
docker-py version: 5.0.0
CPython version: 3.7.10
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
[root@josh josh]#
```

Incase you do not have Docker Compose

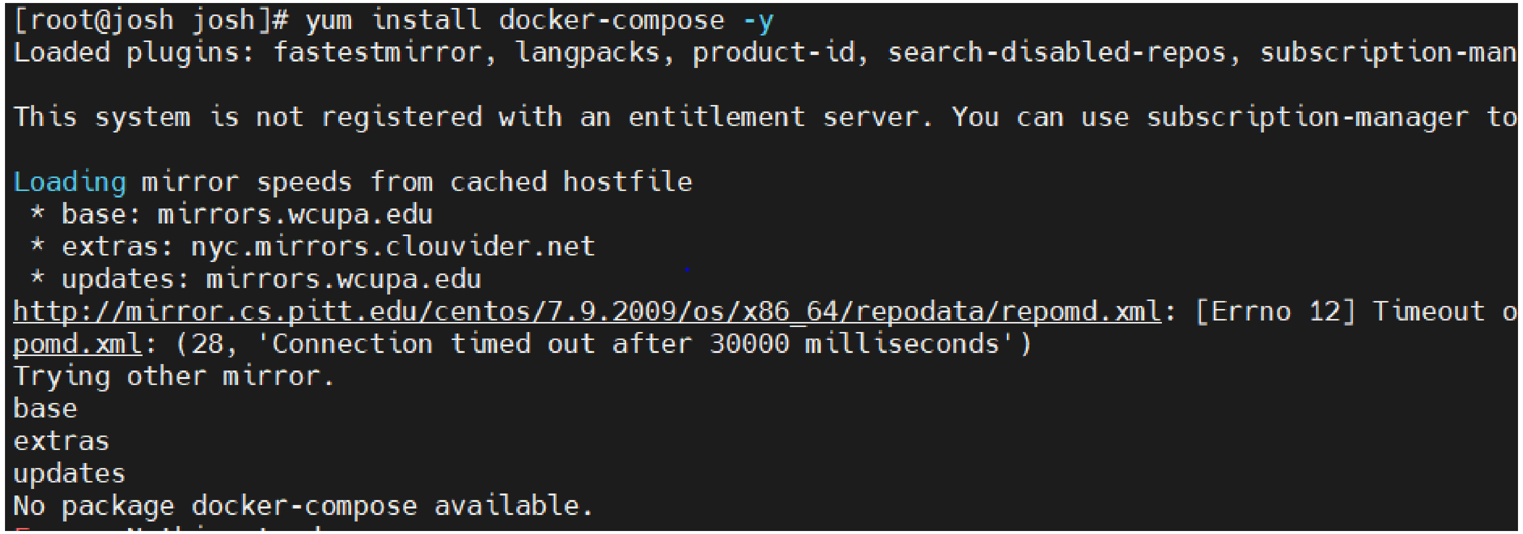
2 Ways to get it

1. https://github.com/docker/compose/releases

2. Using yum install (centos)
   yum install - y docker-compose



- ✓ docker-compose-Darwin-x86_64
- docker-compose-Darwin-x86_64.sha256
- docker-compose-Darwin-x86_64.tgz
- docker-compose-Darwin-x86_64.tgz.sha256
- docker-compose-Linux-x86_64
- docker-compose-Linux-x86_64.sha256



```
[root@josh josh]# yum install docker-compose -y
Loaded plugins: fastestmirror, langpacks, product-id, search-disabled-repos, subscription-man

This system is not registered with an entitlement server. You can use subscription-manager to

Loading mirror speeds from cached hostfile
 * base: mirrors.wcupa.edu
 * extras: nyc.mirrors.clouvider.net
 * updates: mirrors.wcupa.edu
http://mirror.cs.pitt.edu/centos/7.9.2009/os/x86_64/repodata/repomd.xml: [Errno 12] Timeout o
pomd.xml: (28, 'Connection timed out after 30000 milliseconds')
Trying other mirror.
base
extras
updates
No package docker-compose available.
```

Step 2 : Create docker compose file at any location on your system

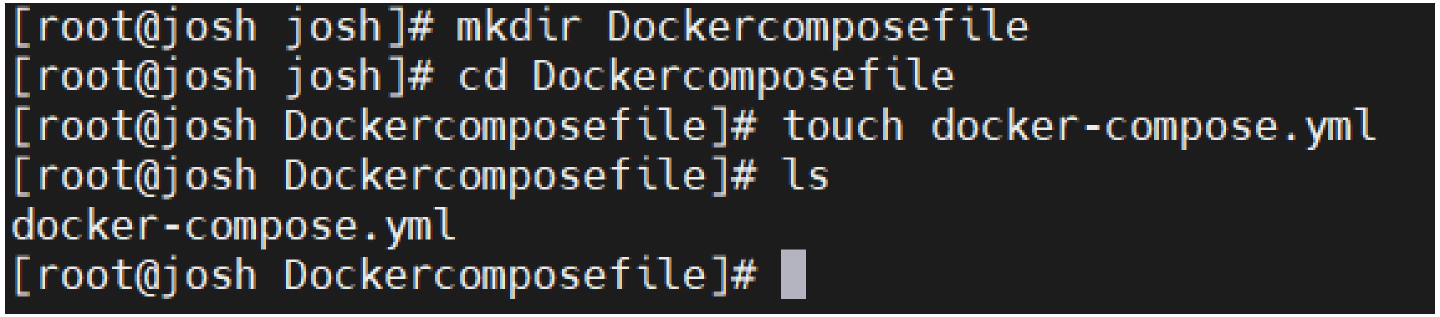
Name of file will be [docker-compose with extension .yml]

docker-compose.yml

Substep 1

Create a folder

```
# mkdir Dockercomposefile
# cd Dockercomposefile
#touch docker-compose.yml
# ls
```



```
[root@josh josh]# mkdir Dockercomposefile
[root@josh josh]# cd Dockercomposefile
[root@josh Dockercomposefile]# touch docker-compose.yml
[root@josh Dockercomposefile]# ls
docker-compose.yml
[root@josh Dockercomposefile]#
```

Use vi or vim editors to put content docker-compose.yml

What do we want to establish

A Web Application

To do this, we need the following images:

They are all open sources

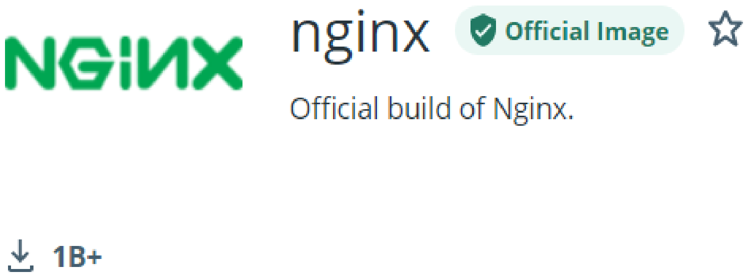
- **A webserver - Nginx:latest or Apache HTTPd:latest**
- **A database image - Mysql or Redis**

I prefer Redis to Mysql because it has fast memory and is more efficient.

Use vi or vim editors to put content docker-compose.yml

- **A webserver - Nginx:latest or Apache HTTPd:latest**
- **A database image - Mysql or Redis**

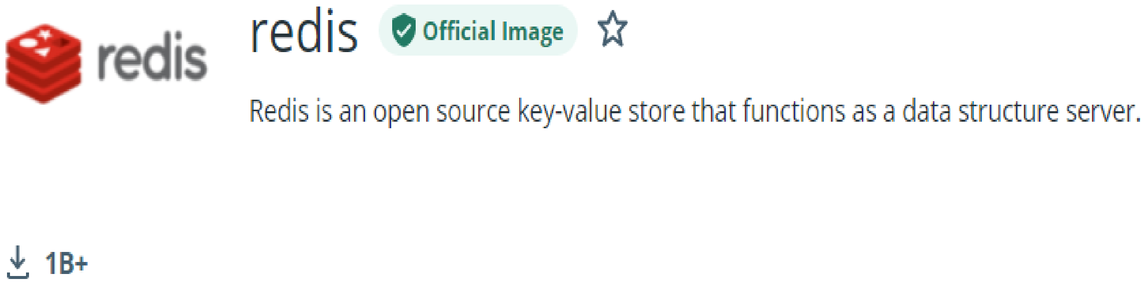
I prefer Redis to Mysql because it has fast memory and is more efficient.



nginx Official Image

Official build of Nginx.

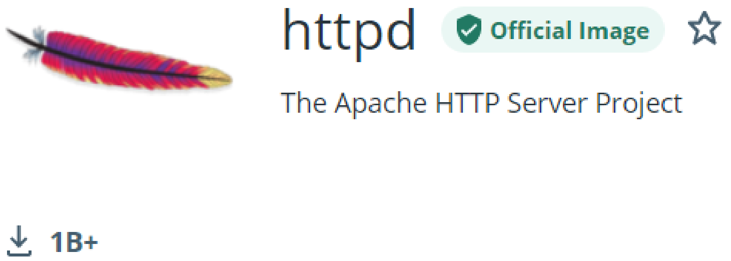
1B+



redis Official Image

Redis is an open source key-value store that functions as a data structure server.

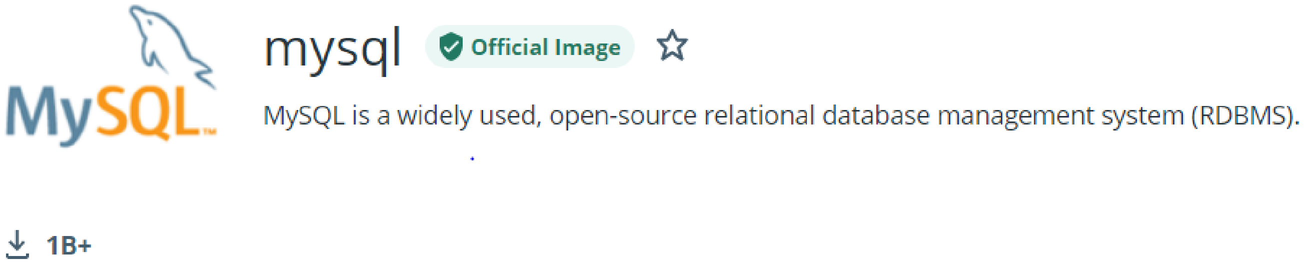
1B+



httpd Official Image

The Apache HTTP Server Project

1B+

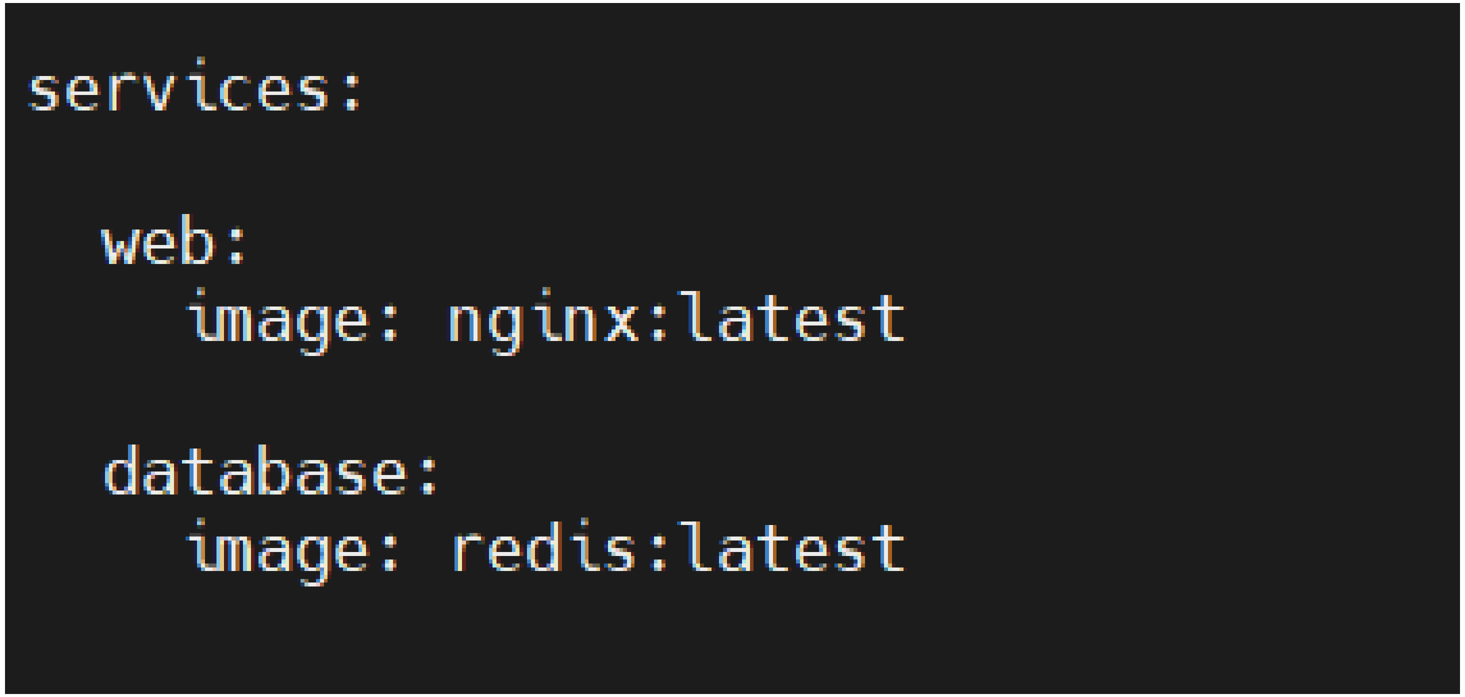


mysql Official Image

MySQL is a widely used, open-source relational database management system (RDBMS).

1B+

A Simple Web Application: [nginx and redis images]



```
services:

  web:
    image: nginx:latest

  database:
    image: redis:latest
```

Rule of Thumb:

Web – use two spaces from the margin
Database – use two spaces from the margin
Image: nginx – use four spaces from margin
Image: redis – use four spaces from margin

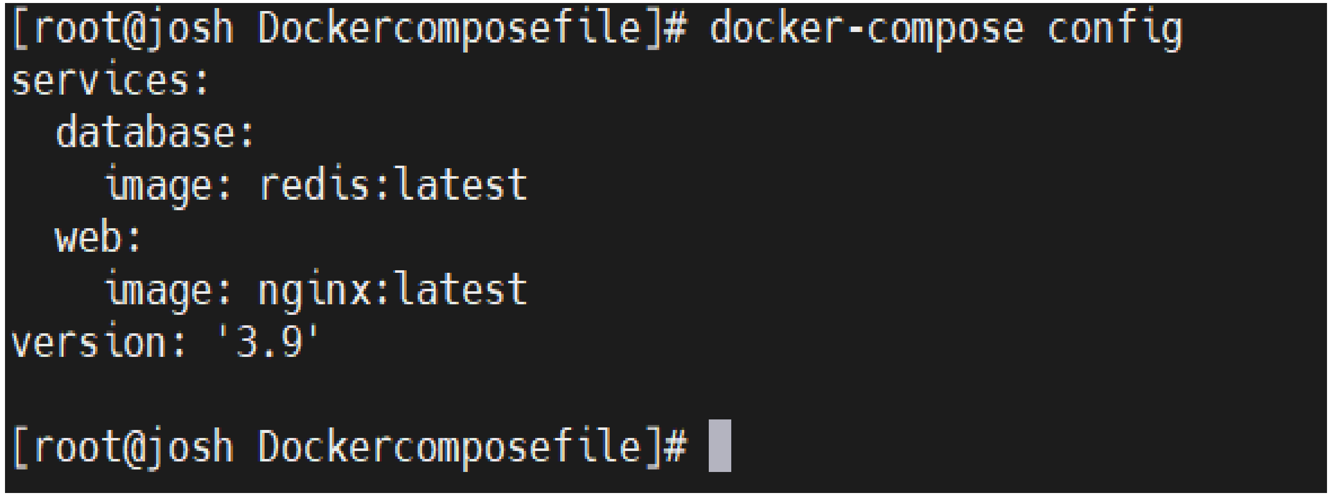
A Simple Web Application: [nginx and redis images]

Step 3 : Check the validity of file by command

Run the command:

docker-compose config

You should not get any error



```
[root@josh Dockercomposefile]# docker-compose config
services:
  database:
    image: redis:latest
  web:
    image: nginx:latest
version: '3.9'

[root@josh Dockercomposefile]# 
```

A Simple Web Application: [nginx and redis images]

Step 4 : Run docker-compose.yml file by command

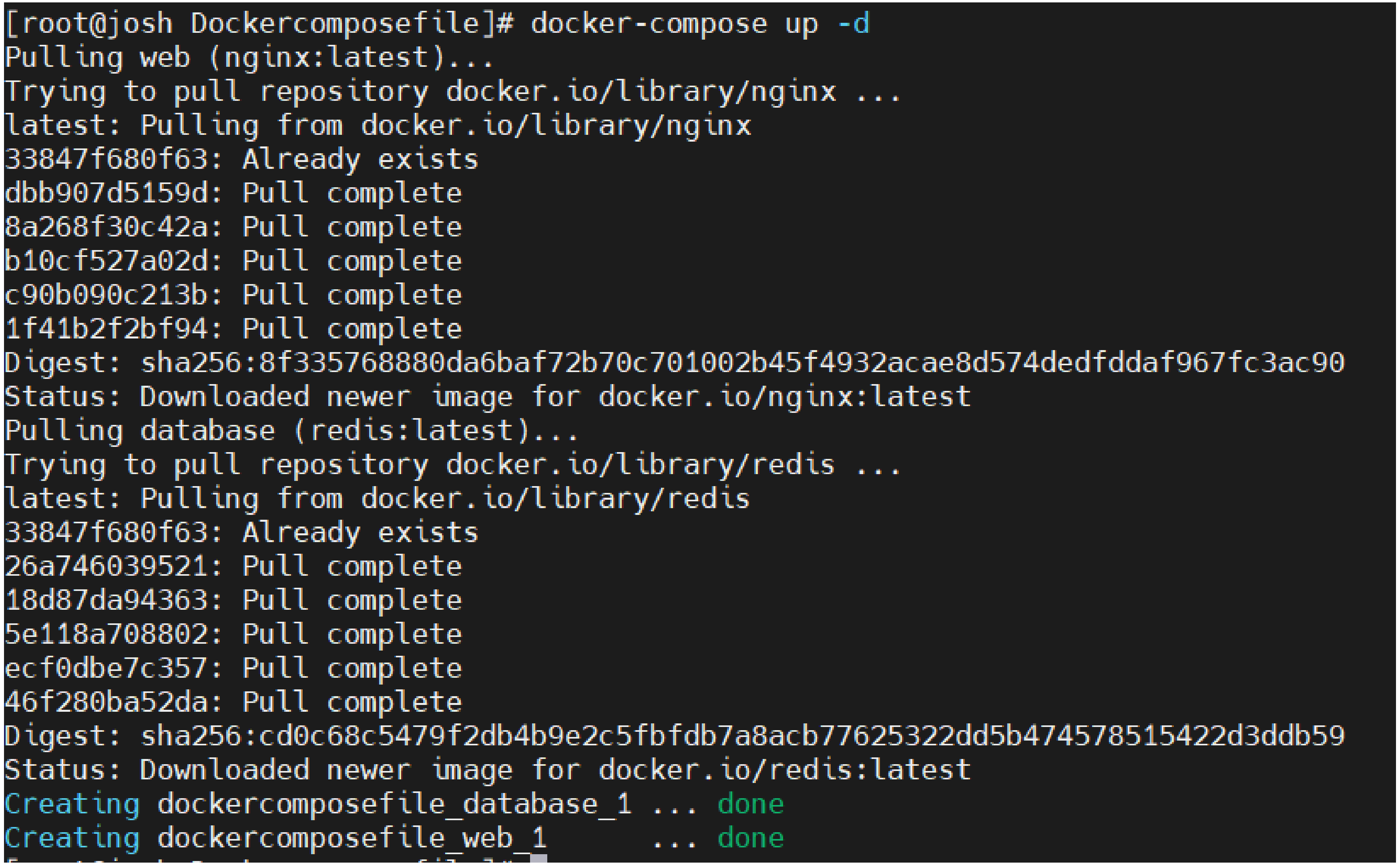
We are going to run the file .yml file we created

Command to use

```
# docker-compose up –d
```

NB: run the file in the detach mode –d

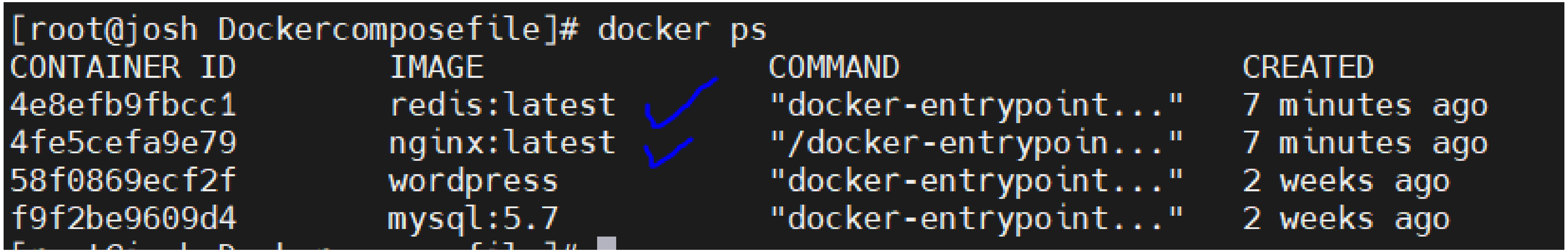
A Simple Web Application: [Web_1 and database_1 Created]



```
[root@josh Dockercomposefile]# docker-compose up -d
Pulling web (nginx:latest)...
Trying to pull repository docker.io/library/nginx ...
latest: Pulling from docker.io/library/nginx
33847f680f63: Already exists
dbb907d5159d: Pull complete
8a268f30c42a: Pull complete
b10cf527a02d: Pull complete
c90b090c213b: Pull complete
1f41b2f2bf94: Pull complete
Digest: sha256:8f335768880da6baf72b70c701002b45f4932acae8d574dedfddaf967fc3ac90
Status: Downloaded newer image for docker.io/nginx:latest
Pulling database (redis:latest)...
Trying to pull repository docker.io/library/redis ...
latest: Pulling from docker.io/library/redis
33847f680f63: Already exists
26a746039521: Pull complete
18d87da94363: Pull complete
5e118a708802: Pull complete
ecf0dbe7c357: Pull complete
46f280ba52da: Pull complete
Digest: sha256:cd0c68c5479f2db4b9e2c5fbfdb7a8acb77625322dd5b474578515422d3ddb59
Status: Downloaded newer image for docker.io/redis:latest
Creating dockercomposefile_database_1 ... done
Creating dockercomposefile_web_1      ... done
```

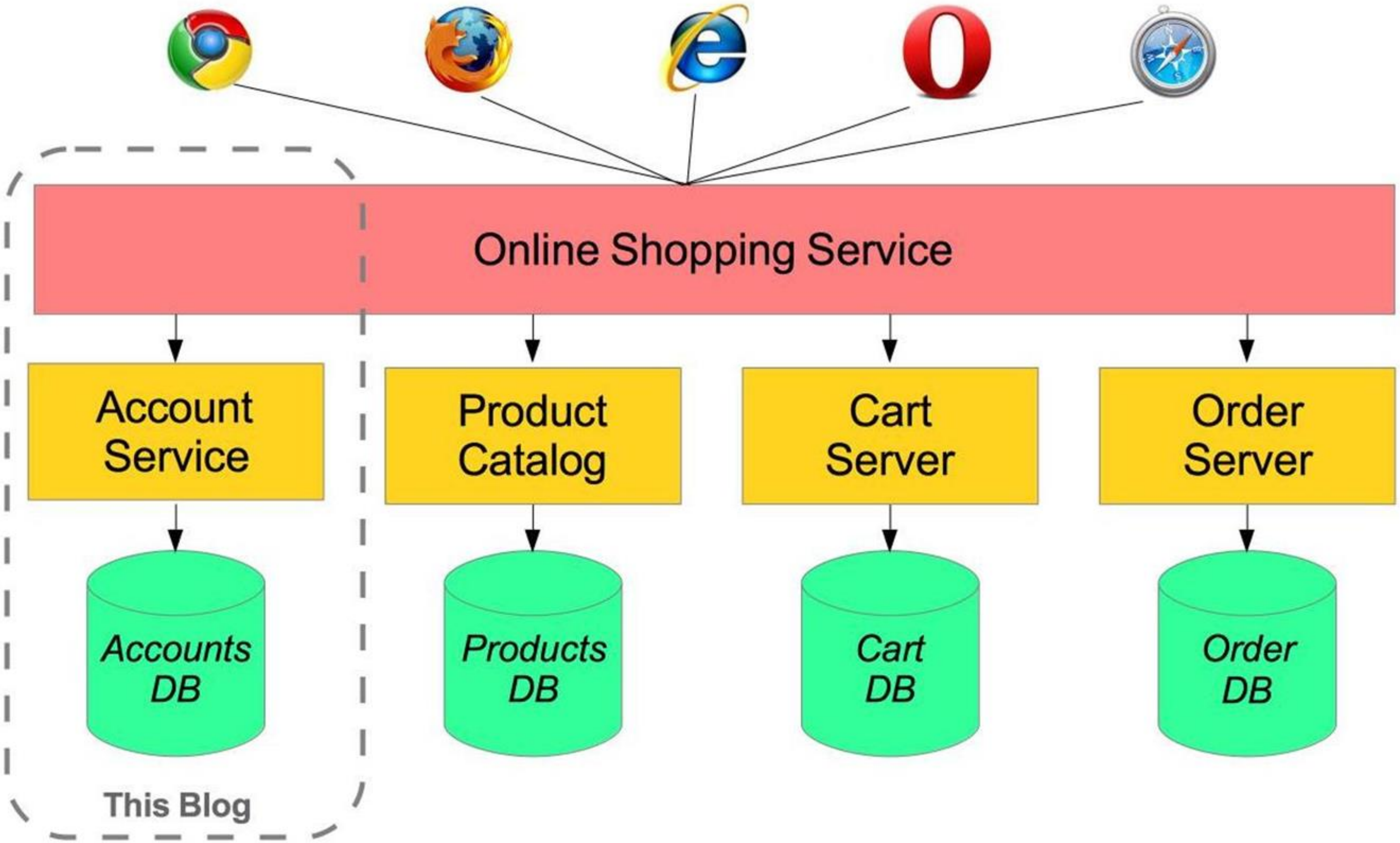
docker-compose up –d [this is the command to run .yml file]

# docker ps [this command will display two containers running with just a single command **{docker-compose up –d}]**



```
[root@josh Dockercomposefile]# docker ps
CONTAINER ID        IMAGE                   COMMAND                   CREATED
4e8efb9fbcc1        redis:latest    ✓       "docker-entrypoint..."    7 minutes ago
4fe5cefa9e79        nginx:latest    ✓       "/docker-entrypoin..."    7 minutes ago
58f0869ecf2f        wordpress               "docker-entrypoint..."    2 weeks ago
f9f2be9609d4        mysql:5.7               "docker-entrypoint..."    2 weeks ago
[root@josh Dockercomposefile]#
```

ENTERPRISE SCALING – AMAZON DURING HIGH SEASONS



ENTERPRISE SCALING – AMAZON DURING HIGH SEASONS

Let assume the product catalog module is running on a Web Application with

- **A webserver - Nginx:latest**
- **A database image - Redis**

During high seasons, especially on thanksgiving, AMAZON.COM has the highest sales volume in USA.

The product catalog manager wants to take care of the increased activities effortlessly by scaling the product catalog module.

ENTERPRISE SCALING – AMAZON DURING HIGH SEASONS

Let assume the product catalog module is running on a Web Application with

- **A webserver - Nginx:latest**
- **A database image - Redis**

During high seasons, especially on thanksgiving, AMAZON.COM has the highest sales volume in USA.

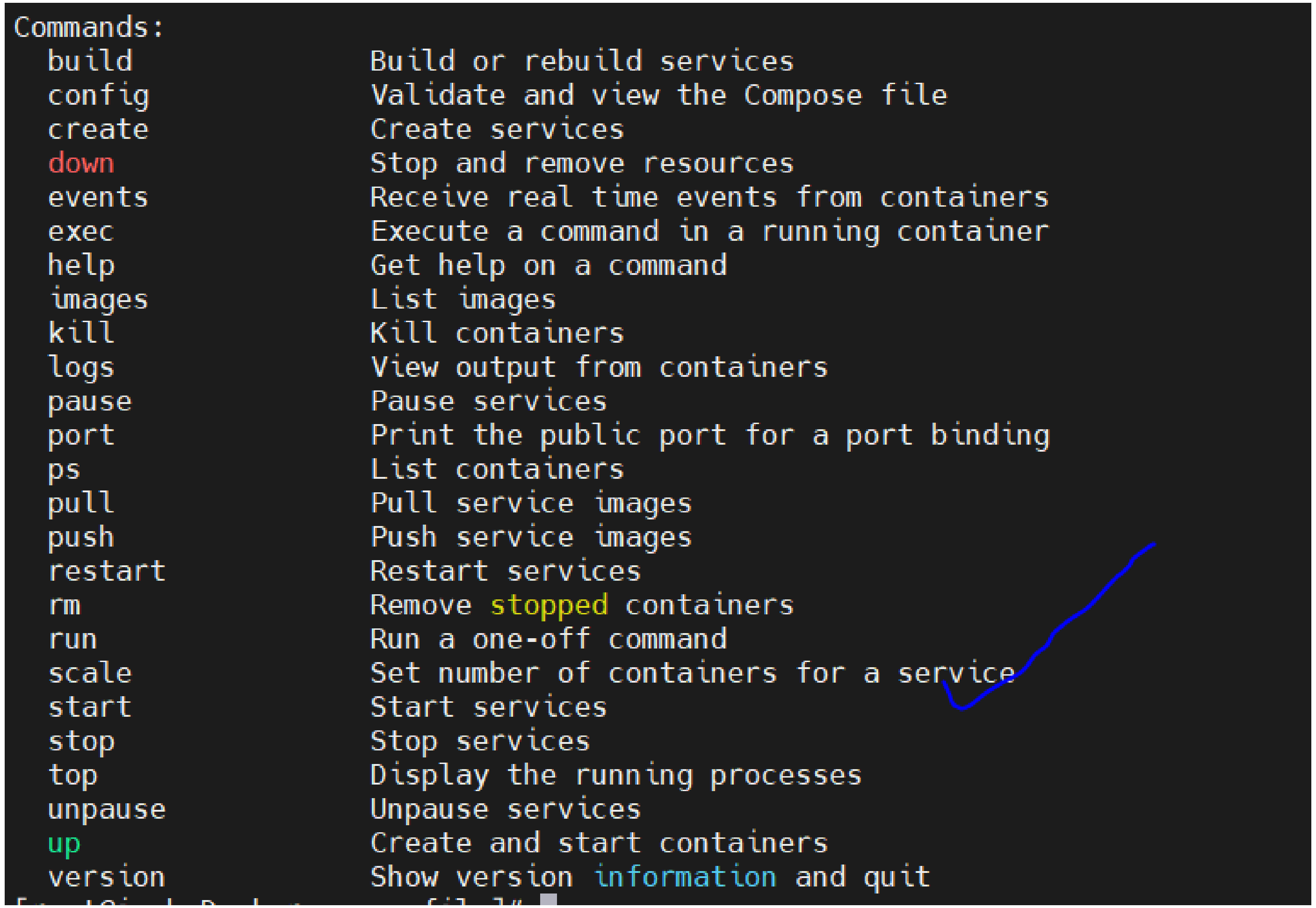
The product catalog manager wants to take care of the increased activities effortlessly by scaling the product catalog module.

HOW TO SCALE SERVICES IN THE ENTERPRISE

TIPS

—scale

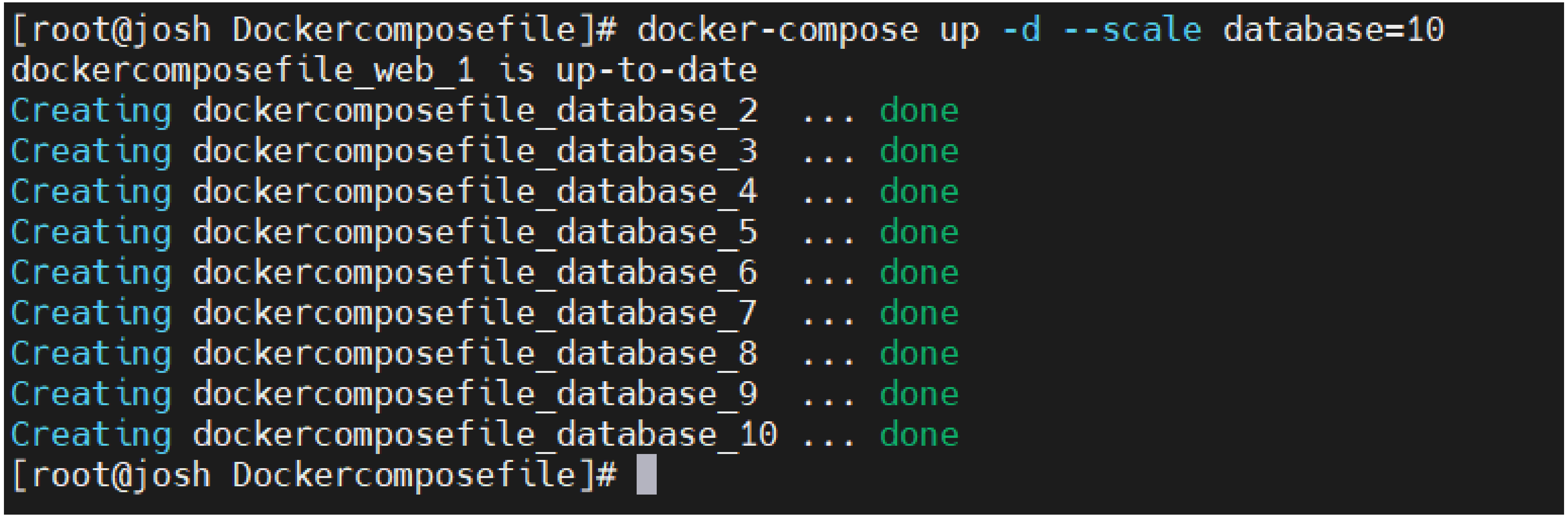
Run the command # docker-compose –help



```
Commands:
  build              Build or rebuild services
  config             Validate and view the Compose file
  create             Create services
  down               Stop and remove resources
  events             Receive real time events from containers
  exec               Execute a command in a running container
  help               Get help on a command
  images             List images
  kill               Kill containers
  logs               View output from containers
  pause              Pause services
  port               Print the public port for a port binding
  ps                 List containers
  pull               Pull service images
  push               Push service images
  restart            Restart services
  rm                 Remove stopped containers
  run                Run a one-off command
  scale              Set number of containers for a service
  start              Start services
  stop               Stop services
  top                Display the running processes
  unpause            Unpause services
  up                 Create and start containers
  version            Show version information and quit
```

HOW TO SCALE SERVICES IN THE ENTERPRISE

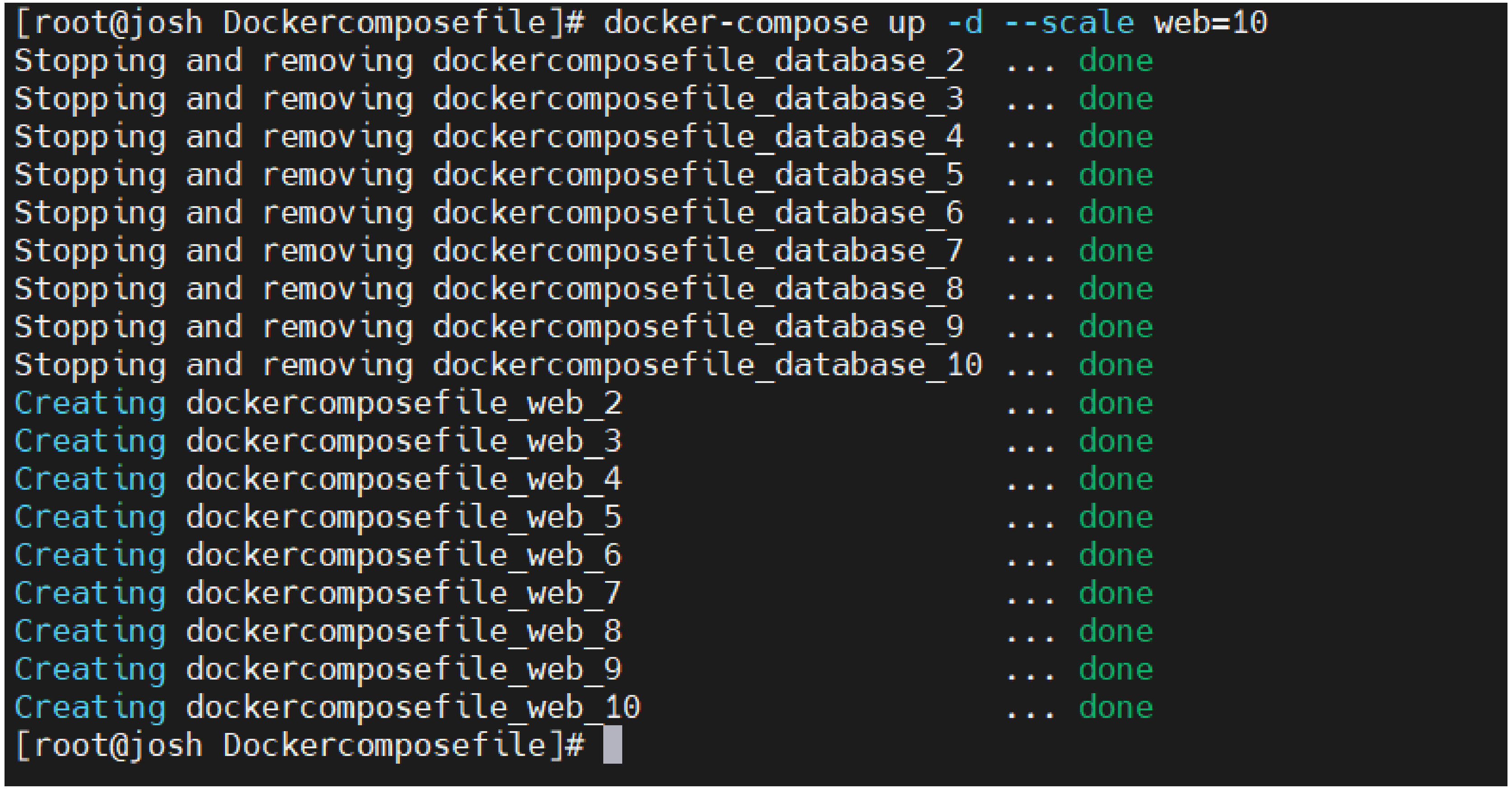
Run the command # docker-compose up -d --scale database=10



```
[root@josh Dockercomposefile]# docker-compose up -d --scale database=10
dockercomposefile_web_1 is up-to-date
Creating dockercomposefile_database_2  ... done
Creating dockercomposefile_database_3  ... done
Creating dockercomposefile_database_4  ... done
Creating dockercomposefile_database_5  ... done
Creating dockercomposefile_database_6  ... done
Creating dockercomposefile_database_7  ... done
Creating dockercomposefile_database_8  ... done
Creating dockercomposefile_database_9  ... done
Creating dockercomposefile_database_10 ... done
[root@josh Dockercomposefile]# 
```

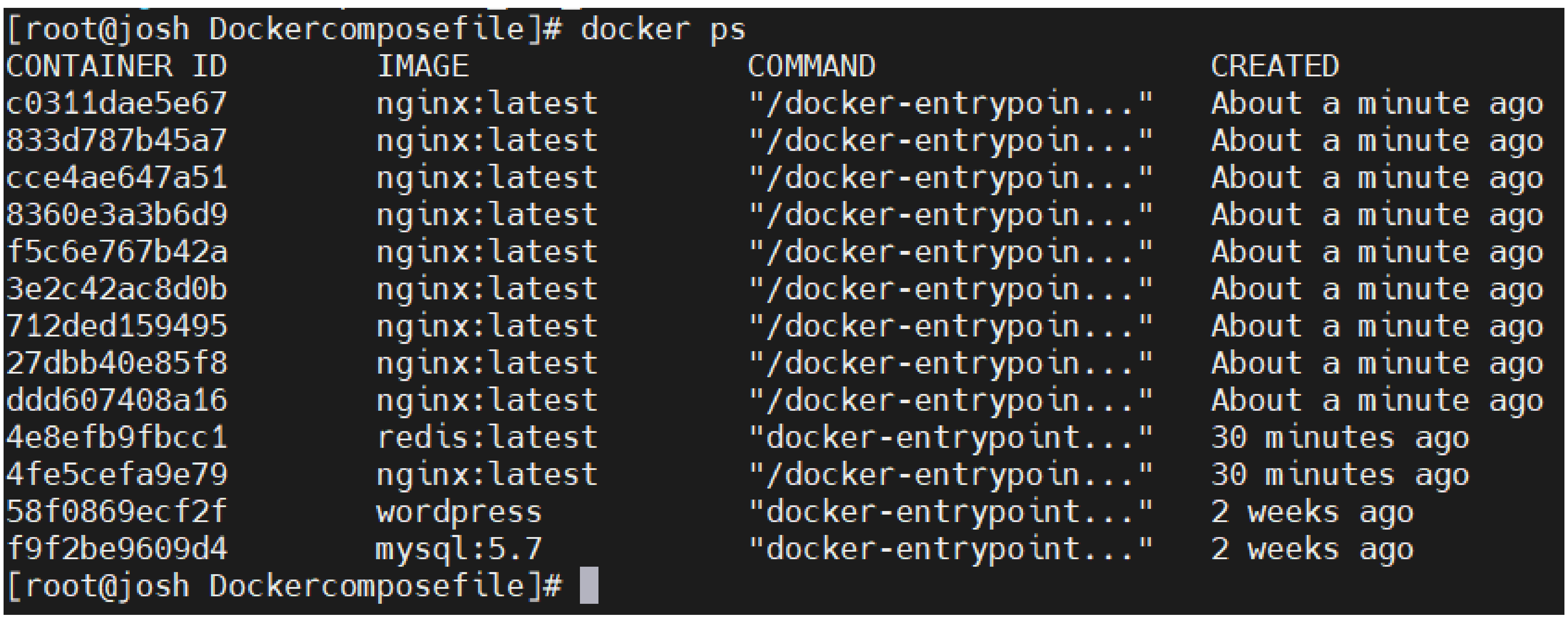
HOW TO SCALE SERVICES IN THE ENTERPRISE

Run the command # docker-compose up -d --scale web=10



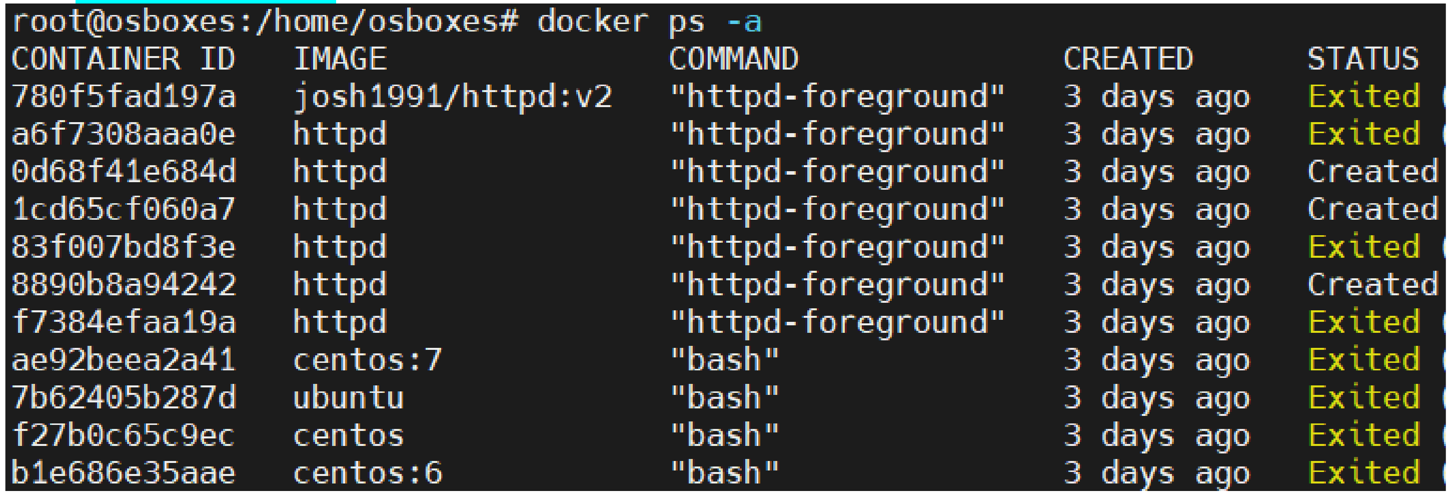
```
[root@josh Dockercomposefile]# docker-compose up -d --scale web=10
Stopping and removing dockercomposefile_database_2  ... done
Stopping and removing dockercomposefile_database_3  ... done
Stopping and removing dockercomposefile_database_4  ... done
Stopping and removing dockercomposefile_database_5  ... done
Stopping and removing dockercomposefile_database_6  ... done
Stopping and removing dockercomposefile_database_7  ... done
Stopping and removing dockercomposefile_database_8  ... done
Stopping and removing dockercomposefile_database_9  ... done
Stopping and removing dockercomposefile_database_10 ... done
Creating dockercomposefile_web_2                    ... done
Creating dockercomposefile_web_3                    ... done
Creating dockercomposefile_web_4                    ... done
Creating dockercomposefile_web_5                    ... done
Creating dockercomposefile_web_6                    ... done
Creating dockercomposefile_web_7                    ... done
Creating dockercomposefile_web_8                    ... done
Creating dockercomposefile_web_9                    ... done
Creating dockercomposefile_web_10                   ... done
[root@josh Dockercomposefile]#
```

HOW TO SCALE SERVICES IN THE ENTERPRISE



```
[root@josh Dockercomposefile]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
c0311dae5e67        nginx:latest        "/docker-entrypoin..."   About a minute ago
833d787b45a7        nginx:latest        "/docker-entrypoin..."   About a minute ago
cce4ae647a51        nginx:latest        "/docker-entrypoin..."   About a minute ago
8360e3a3b6d9        nginx:latest        "/docker-entrypoin..."   About a minute ago
f5c6e767b42a        nginx:latest        "/docker-entrypoin..."   About a minute ago
3e2c42ac8d0b        nginx:latest        "/docker-entrypoin..."   About a minute ago
712ded159495        nginx:latest        "/docker-entrypoin..."   About a minute ago
27dbb40e85f8        nginx:latest        "/docker-entrypoin..."   About a minute ago
ddd607408a16        nginx:latest        "/docker-entrypoin..."   About a minute ago
4e8efb9fbcc1        redis:latest        "docker-entrypoint..."   30 minutes ago
4fe5cefa9e79        nginx:latest        "/docker-entrypoin..."   30 minutes ago
58f0869ecf2f        wordpress           "docker-entrypoint..."   2 weeks ago
f9f2be9609d4        mysql:5.7           "docker-entrypoint..."   2 weeks ago
[root@josh Dockercomposefile]#
```

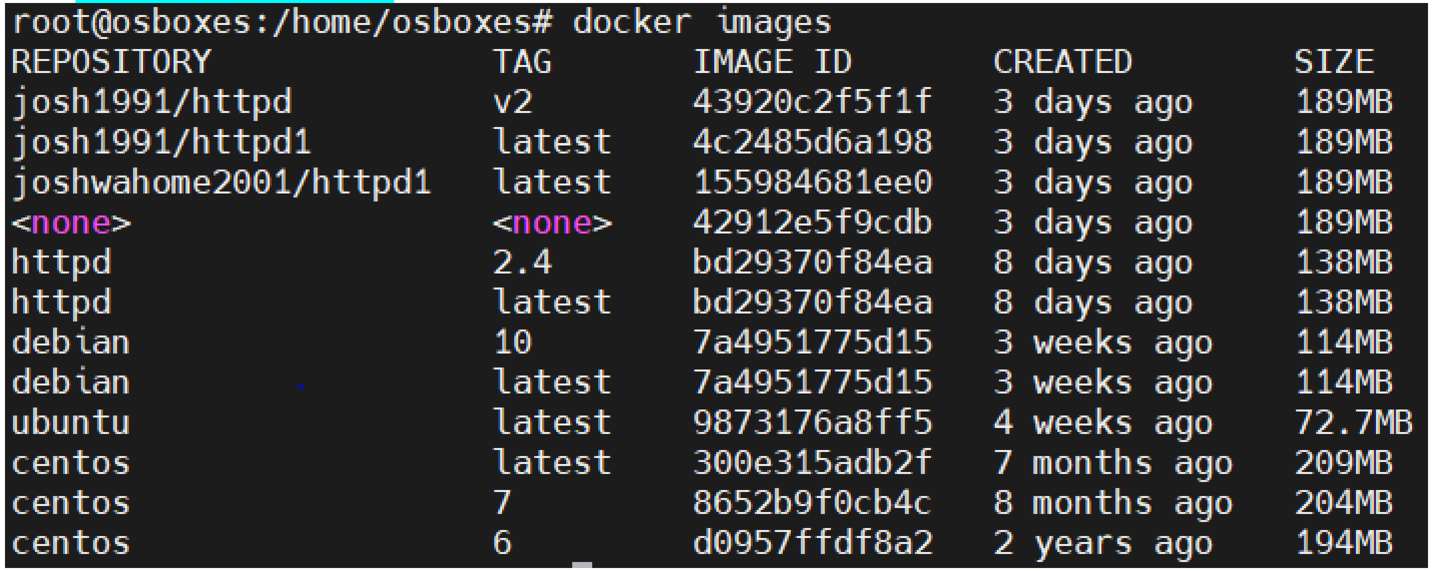
10. # docker ps -a



```
root@osboxes:/home/osboxes# docker ps -a
CONTAINER ID   IMAGE              COMMAND                CREATED      STATUS
780f5fad197a   josh1991/httpd:v2  "httpd-foreground"     3 days ago   Exited (
a6f7308aaa0e   httpd              "httpd-foreground"     3 days ago   Exited (
0d68f41e684d   httpd              "httpd-foreground"     3 days ago   Created
1cd65cf060a7   httpd              "httpd-foreground"     3 days ago   Created
83f007bd8f3e   httpd              "httpd-foreground"     3 days ago   Exited (
8890b8a94242   httpd              "httpd-foreground"     3 days ago   Created
f7384efaa19a   httpd              "httpd-foreground"     3 days ago   Exited (
ae92beea2a41   centos:7           "bash"                 3 days ago   Exited (
7b62405b287d   ubuntu             "bash"                 3 days ago   Exited (
f27b0c65c9ec   centos             "bash"                 3 days ago   Exited (
b1e686e35aae   centos:6           "bash"                 3 days ago   Exited (
```

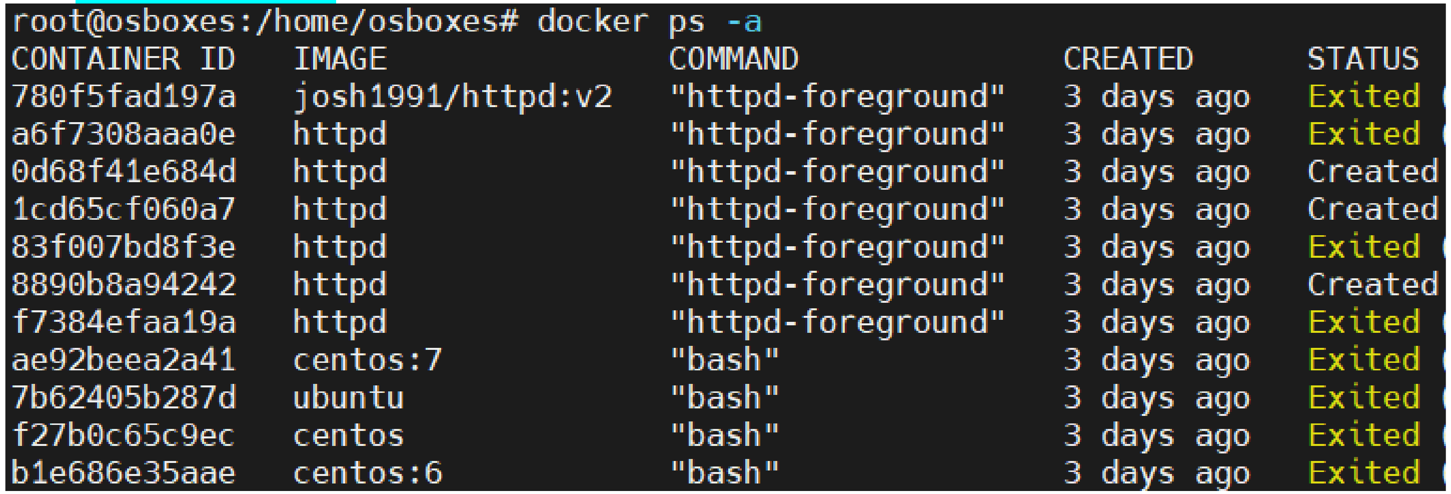
Show both the running and exited container.

11. # docker images



```
root@osboxes:/home/osboxes# docker images
REPOSITORY              TAG       IMAGE ID       CREATED        SIZE
josh1991/httpd          v2        43920c2f5f1f   3 days ago     189MB
josh1991/httpd1         latest    4c2485d6a198   3 days ago     189MB
joshwahome2001/httpd1   latest    155984681ee0   3 days ago     189MB
<none>                  <none>    42912e5f9cdb   3 days ago     189MB
httpd                   2.4       bd29370f84ea   8 days ago     138MB
httpd                   latest    bd29370f84ea   8 days ago     138MB
debian                  10        7a4951775d15   3 weeks ago    114MB
debian                  latest    7a4951775d15   3 weeks ago    114MB
ubuntu                  latest    9873176a8ff5   4 weeks ago    72.7MB
centos                  latest    300e315adb2f   7 months ago   209MB
centos                  7         8652b9f0cb4c   8 months ago   204MB
centos                  6         d0957ffdf8a2   2 years ago    194MB
```

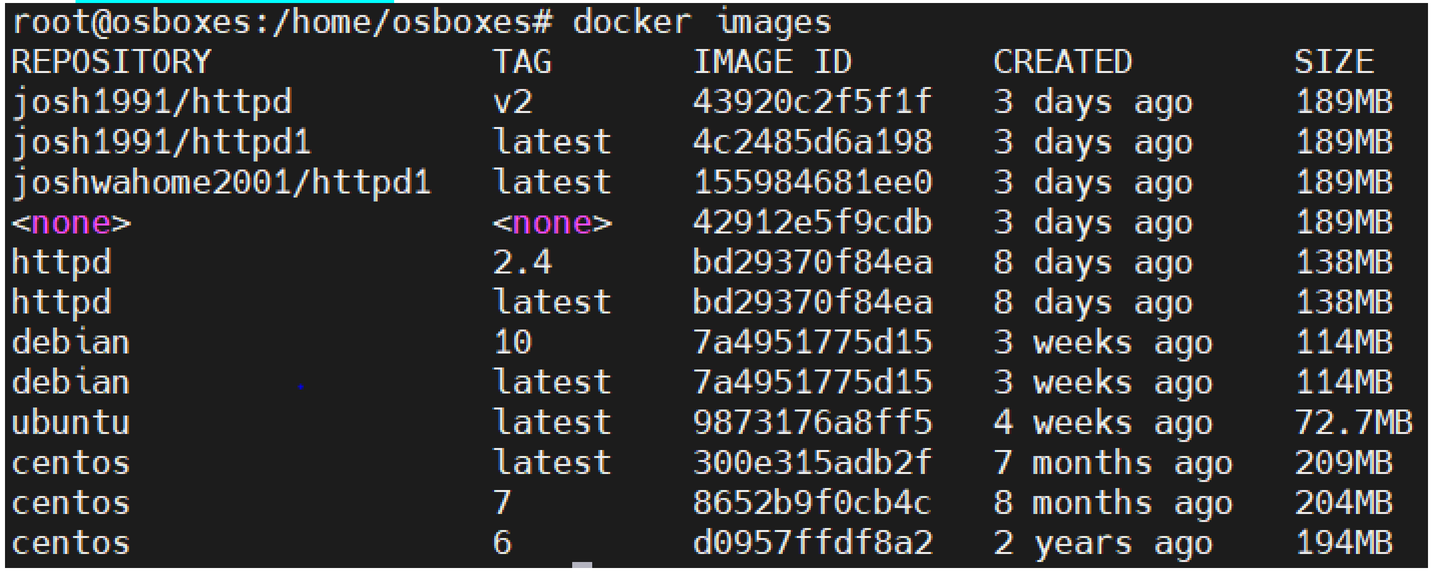
10. # docker ps -a



```
root@osboxes:/home/osboxes# docker ps -a
CONTAINER ID   IMAGE              COMMAND                CREATED       STATUS
780f5fad197a   josh1991/httpd:v2  "httpd-foreground"     3 days ago    Exited (
a6f7308aaa0e   httpd              "httpd-foreground"     3 days ago    Exited (
0d68f41e684d   httpd              "httpd-foreground"     3 days ago    Created
1cd65cf060a7   httpd              "httpd-foreground"     3 days ago    Created
83f007bd8f3e   httpd              "httpd-foreground"     3 days ago    Exited (
8890b8a94242   httpd              "httpd-foreground"     3 days ago    Created
f7384efaa19a   httpd              "httpd-foreground"     3 days ago    Exited (
ae92beea2a41   centos:7           "bash"                 3 days ago    Exited (
7b62405b287d   ubuntu             "bash"                 3 days ago    Exited (
f27b0c65c9ec   centos             "bash"                 3 days ago    Exited (
b1e686e35aae   centos:6           "bash"                 3 days ago    Exited (
```

Show both the running and exited container.

11. # docker images



```
root@osboxes:/home/osboxes# docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
josh1991/httpd          v2        43920c2f5f1f   3 days ago      189MB
josh1991/httpd1         latest    4c2485d6a198   3 days ago      189MB
joshwahome2001/httpd1   latest    155984681ee0   3 days ago      189MB
<none>                  <none>    42912e5f9cdb   3 days ago      189MB
httpd                   2.4       bd29370f84ea   8 days ago      138MB
httpd                   latest    bd29370f84ea   8 days ago      138MB
debian                  10        7a4951775d15   3 weeks ago     114MB
debian                  latest    7a4951775d15   3 weeks ago     114MB
ubuntu                  latest    9873176a8ff5   4 weeks ago     72.7MB
centos                  latest    300e315adb2f   7 months ago    209MB
centos                  7         8652b9f0cb4c   8 months ago    204MB
centos                  6         d0957ffdf8a2   2 years ago     194MB
```

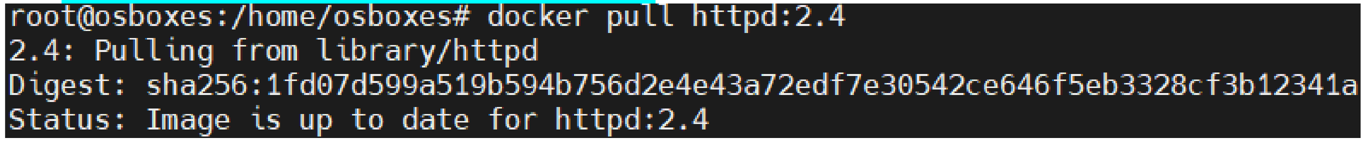
Ubuntu

docker pull ubuntu - if you do not specify, it will pull the latest image

12. # docker pull ubuntu

# docker pull centos:5

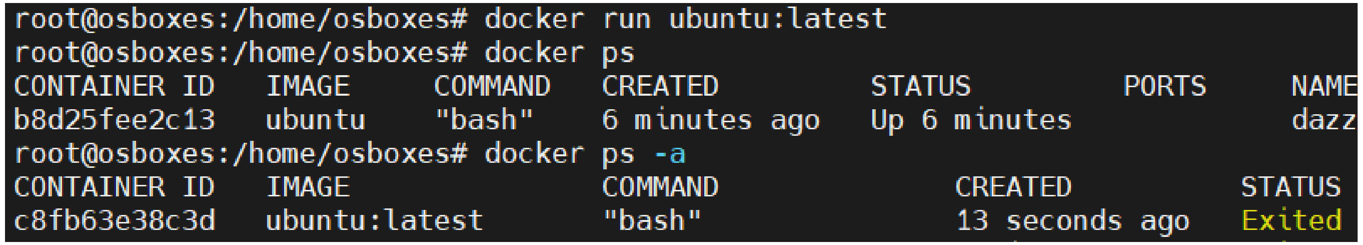
13. # docker pull httpd:2.4



```
root@osboxes:/home/osboxes# docker pull httpd:2.4
2.4: Pulling from library/httpd
Digest: sha256:1fd07d599a519b594b756d2e4e43a72edf7e30542ce646f5eb3328cf3b12341a
Status: Image is up to date for httpd:2.4
```

14. # docker run - will allow you to get stright to the container

example: # docker run ubuntu:latest



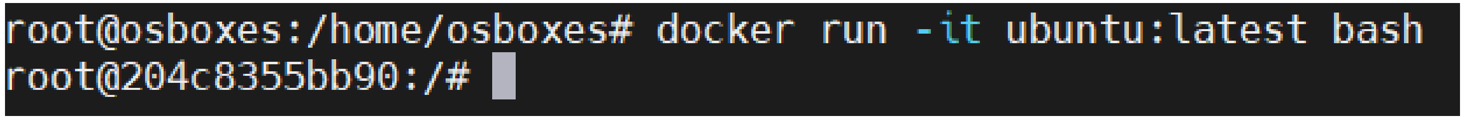
```
root@osboxes:/home/osboxes# docker run ubuntu:latest
root@osboxes:/home/osboxes# docker ps
CONTAINER ID   IMAGE     COMMAND    CREATED          STATUS         PORTS      NAME
b8d25fee2c13   ubuntu    "bash"     6 minutes ago    Up 6 minutes              dazz
root@osboxes:/home/osboxes# docker ps -a
CONTAINER ID   IMAGE             COMMAND              CREATED          STATUS
c8fb63e38c3d   ubuntu:latest     "bash"               13 seconds ago   Exited
```

Docker run but exited

we can solve this by giving the container - it (interactive terminal) and specifying where the shell the of the (bash)

Let recall the command and add the following: # docker run -it ubuntu:latest bash

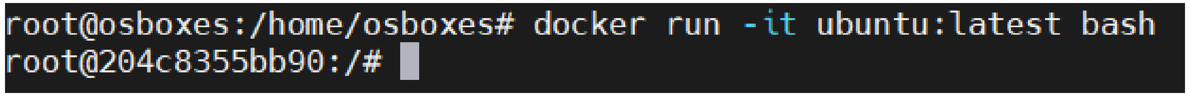
It takes you straight to the container



```
root@osboxes:/home/osboxes# docker run -it ubuntu:latest bash
root@204c8355bb90:/#
```

Exit and run the command # docker ps and # docker ps -a

Let recall the command and add the following: # docker run -it ubuntu:latest bash

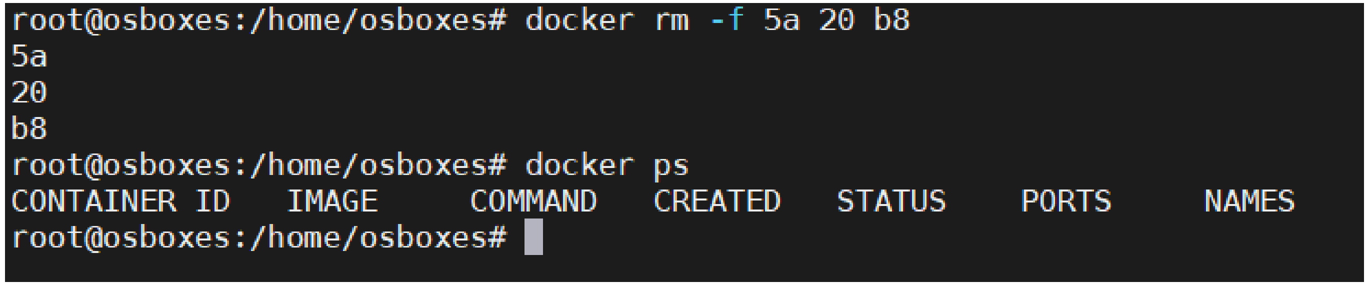


```
root@osboxes:/home/osboxes# docker run -it ubuntu:latest bash
root@204c8355bb90:/# 
```

Ctrl p + q   the ENTER and run the command # docker ps and # docker ps -a

Remove the running container

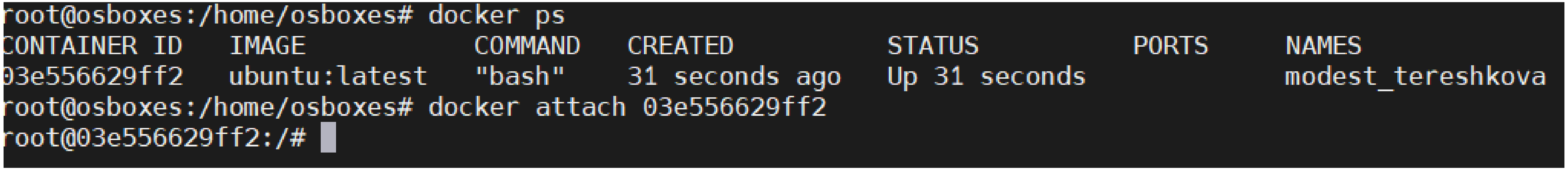
# docker rm -f [container ID]



```
root@osboxes:/home/osboxes# docker rm -f 5a 20 b8
5a
20
b8
root@osboxes:/home/osboxes# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@osboxes:/home/osboxes# 
```

To get back to a running container:

# docker attach [container id]



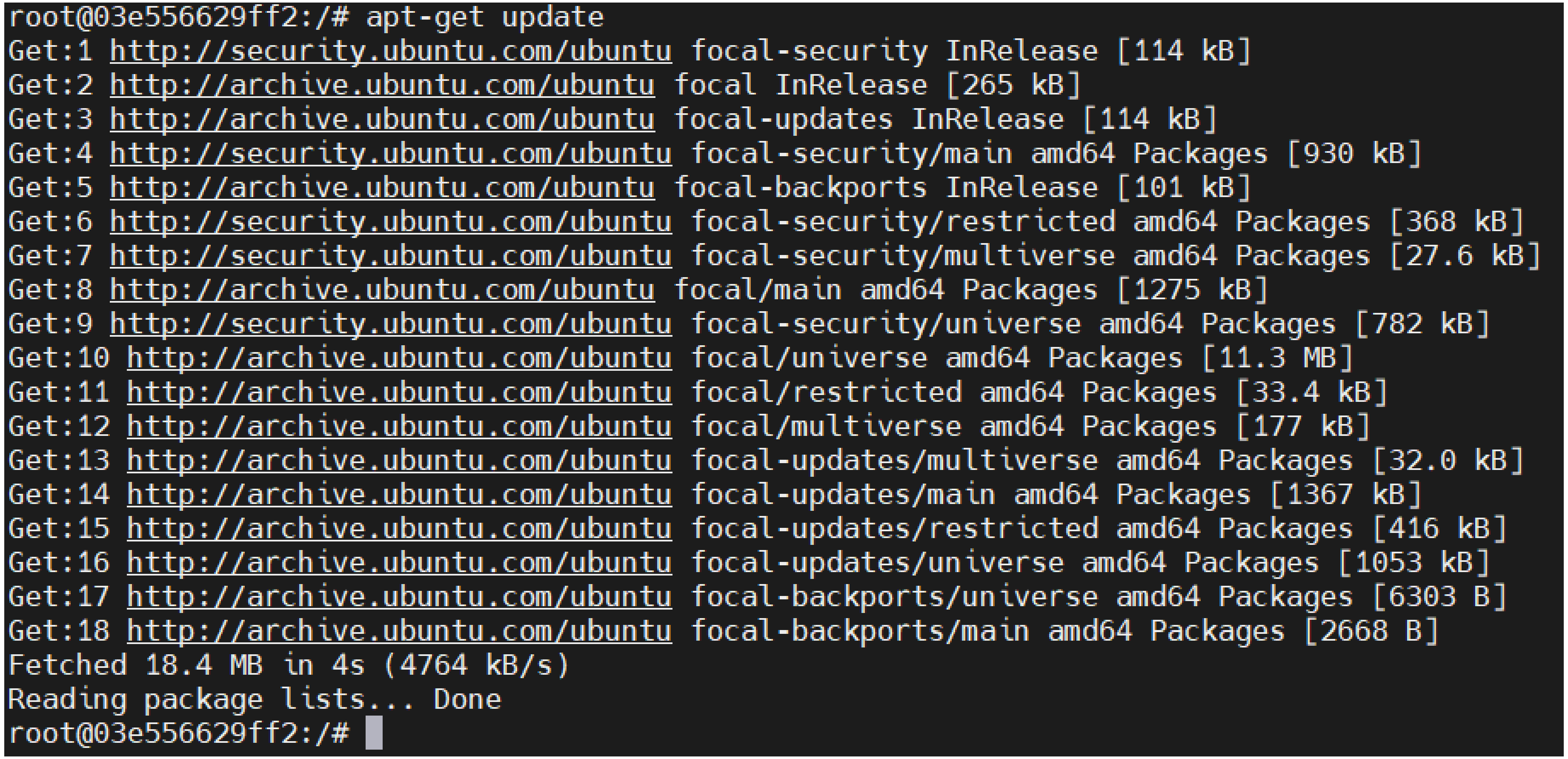
```
root@osboxes:/home/osboxes# docker ps
CONTAINER ID   IMAGE           COMMAND   CREATED          STATUS          PORTS     NAMES
03e556629ff2   ubuntu:latest   "bash"    31 seconds ago   Up 31 seconds             modest_tereshkova
root@osboxes:/home/osboxes# docker attach 03e556629ff2
root@03e556629ff2:/#
```

Since we are inside ubuntu container, let do some updates

# apt-get updates

Since we are inside ubuntu container, let do some updates

# apt-get updates



```
root@03e556629ff2:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [930 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [368 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.6 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [782 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.0 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1367 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [416 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1053 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6303 B]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2668 B]
Fetched 18.4 MB in 4s (4764 kB/s)
Reading package lists... Done
root@03e556629ff2:/#
```

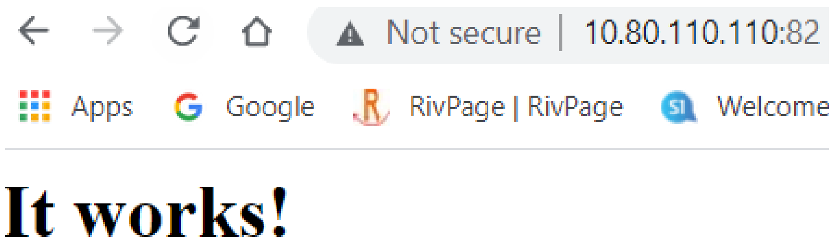
```
# docker run -itd -p 82:80 httpd
```

Do not add the bash shell We want it to be in the detach mode. Othwewise it will not work

Go to ifconfig and get your IP address

Mine is 10:80:110:110:82

Results



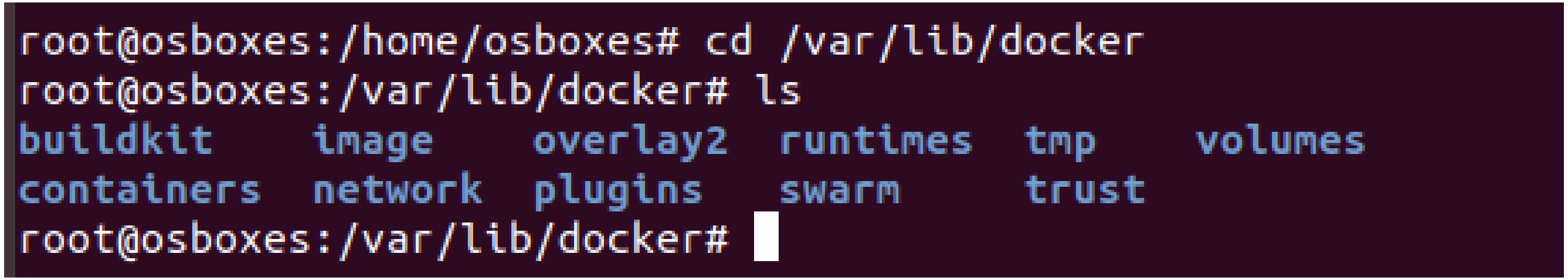
It works!

A docker image will behave the same regardless of the the environment.

When you install docker, a path is created:

var/lib/docker

#cd /var/lib/docker



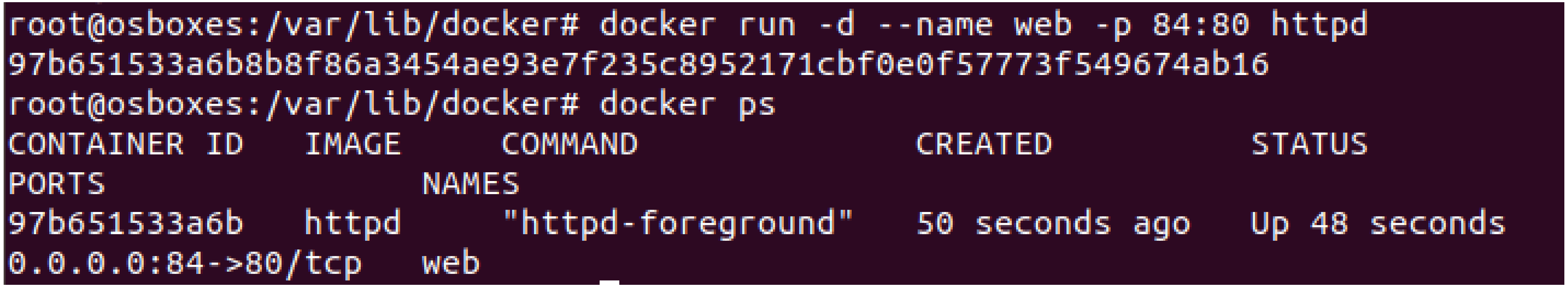
```
root@osboxes:/home/osboxes# cd /var/lib/docker
root@osboxes:/var/lib/docker# ls
buildkit    image    overlay2  runtimes  tmp    volumes
containers  network  plugins   swarm     trust
root@osboxes:/var/lib/docker# 
```

Giving a container a name and port number

# docker run -itd --name web -p 84:80 httpd

-itd means in detach mode - do not put the intercatice shell like bash. It will not work

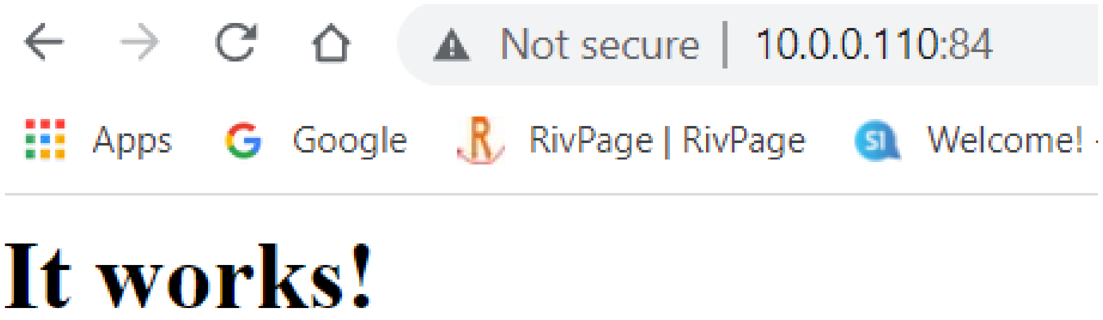
NB - no need to put = sign



```
root@osboxes:/var/lib/docker# docker run -d --name web -p 84:80 httpd
97b651533a6b8b8f86a3454ae93e7f235c8952171cbf0e0f57773f549674ab16
root@osboxes:/var/lib/docker# docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS
PORTS                NAMES
97b651533a6b   httpd     "httpd-foreground"   50 seconds ago   Up 48 seconds
0.0.0.0:84->80/tcp   web
```

Again let run the IP address on the web

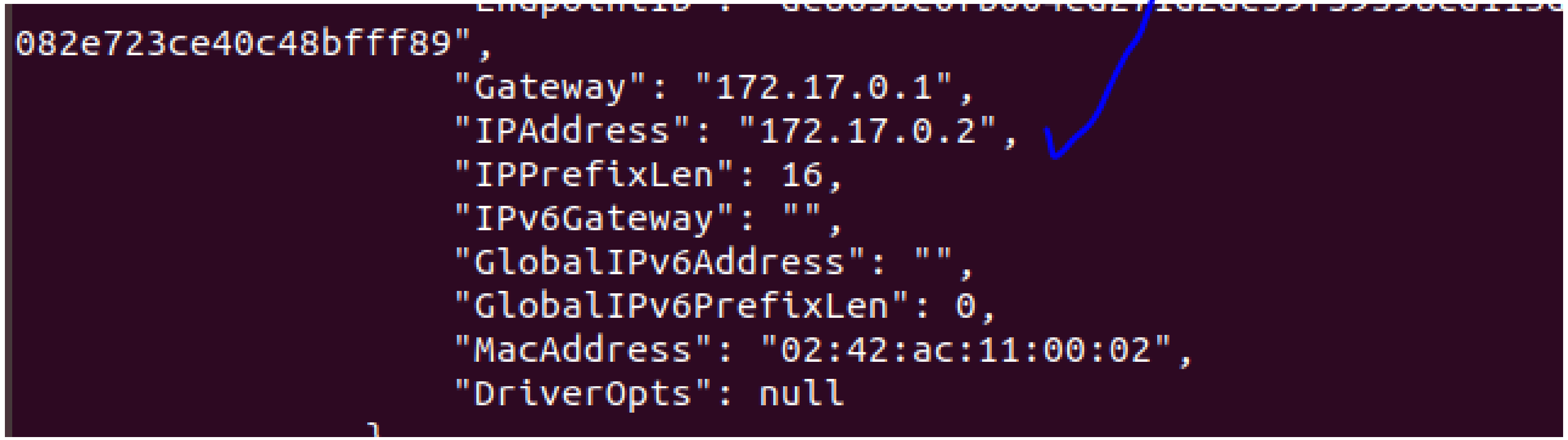
Mine:10.0.0.110:84



It works!

What if we want our own information to display instead of IT WORKS

Run # docker inspect [container ID]



```
082e723ce40c48bfff89",
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:02",
                    "DriverOpts": null
```

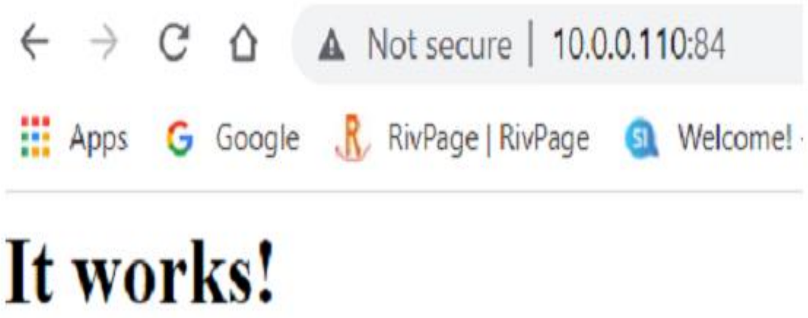
Make sure you are in the root directory. You do not want to mess with anything in the /var/lib/docker

The conatiner ip is: 172.17.0.2

volume - Mounting a container

Again let run the IP address on the web

Mine:10.0.0.110:84



It works!

Make sure you are in the home directory [root directory might not work well]

it works! is using the path : /usr/local/apache2/htdocs/

```
-v /home/html : /usr/local/apache2/htdocs/
```

-v is for mounting the volume

Volume is the ability to create a host folder (HTML) that will mount on the container folder (HTDOCS) and save the information incase we lose the container.

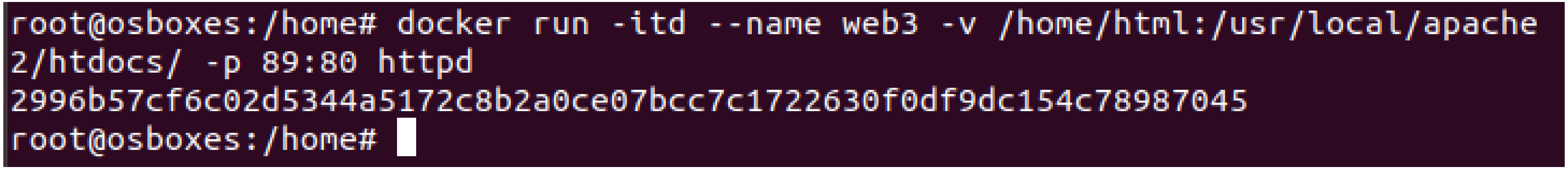
Run the command

```
# docker run -itd --name web 2 -v /home/html : /usr/local/apache2/htdocs/ -p 87:80 httpd
```

volume - Mounting a container

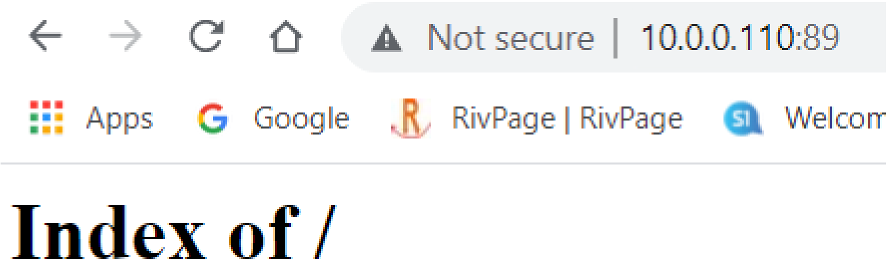
Run the command

```
# docker run -itd --name web 3 -v /home/html :
/usr/local/apache2/htdocs/   -p 89:80 httpd
```



```
root@osboxes:/home# docker run -itd --name web3 -v /home/html:/usr/local/apache
2/htdocs/ -p 89:80 httpd
2996b57cf6c02d5344a5172c8b2a0ce07bcc7c1722630f0df9dc154c78987045
root@osboxes:/home# 
```

Run your IP Address on a web server: 10.0.0.110:89



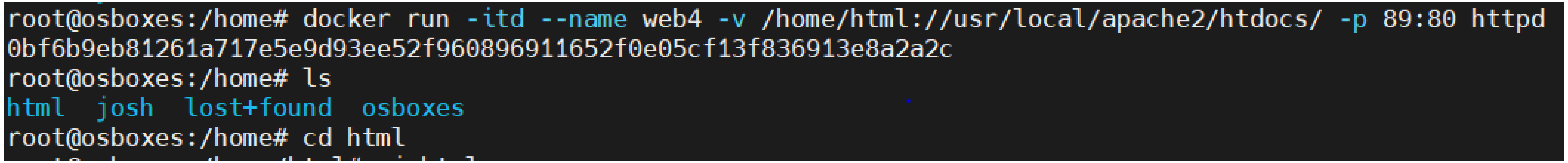
Index of /

volume - Mounting a container

Let us put information on the container and display the information on the host

Run the command

```
# docker run -itd --name web 5 -v /home/html://usr/local/apache2/htdocs/  -p 89:80 httpd
```



```
root@osboxes:/home# docker run -itd --name web4 -v /home/html://usr/local/apache2/htdocs/ -p 89:80 httpd
0bf6b9eb81261a717e5e9d93ee52f960896911652f0e05cf13f836913e8a2a2c
root@osboxes:/home# ls
html  josh  lost+found  osboxes
root@osboxes:/home# cd html
```

Home directory

```
# ls
```

```
# vi html
```

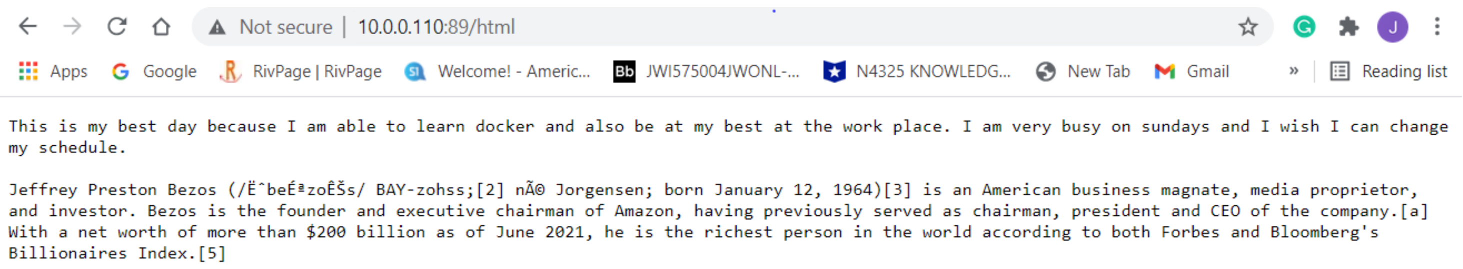
put the following content from this link

https://en.wikipedia.org/wiki/Jeff_Bezos

```
:wq
```

Go on your webserver and type the ip address: Mine is 10.0.0.110:89

Results



```
This is my best day because I am able to learn docker and also be at my best at the work place. I am very busy on sundays and I wish I can change
my schedule.

Jeffrey Preston Bezos (/Ë̂beÉªzoÊŠs/ BAY-zohss;[2] nÃ© Jorgensen; born January 12, 1964)[3] is an American business magnate, media proprietor,
and investor. Bezos is the founder and executive chairman of Amazon, having previously served as chairman, president and CEO of the company.[a]
With a net worth of more than $200 billion as of June 2021, he is the richest person in the world according to both Forbes and Bloomberg's
Billionaires Index.[5]
```

Important Information

You can create many containers using different ports but mounted on the same volume

For example: 89:80 87:80 86:80 85:80 84:80 83:80 etc

```
#docker run -itd --name web 6 -v /home/html://usr/local/apache2/htdocs/ -p 89:80 httpd
```

A link between html and htdocs was created. The content you put on the html folder will also go to the htdocs folder in the container

How to Build our own Image

Create a file called Dockerfile

```
# touch Dockerfile
```

Very Important:

Create INDEX.HTML file

```
#vim index.html
```

Put this content

```
<h1> this is my images for httpd </h1>
:wq
```

```
# vim Dockerfile

FROM httpd:2.4      [take instructions from]

MAINTAINER [put your email or your name]

RUN  apt-get update    [update all the information]

COPY   ./index.html /usr/local/apache2/htdocs/

:wq
```

DO NOT FORGET THE DOT infront of /index.html/ file

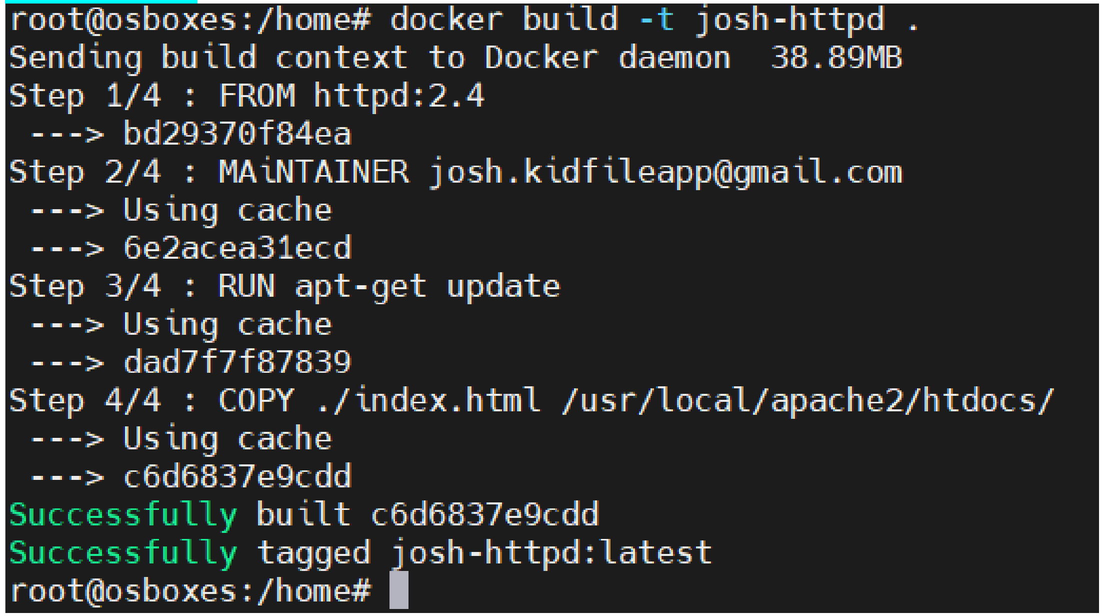
Run the build command

# docker build -t josh-httpd .

The dot is import because it says [Dockerfile is right here]

make sure you are logged to your docker account

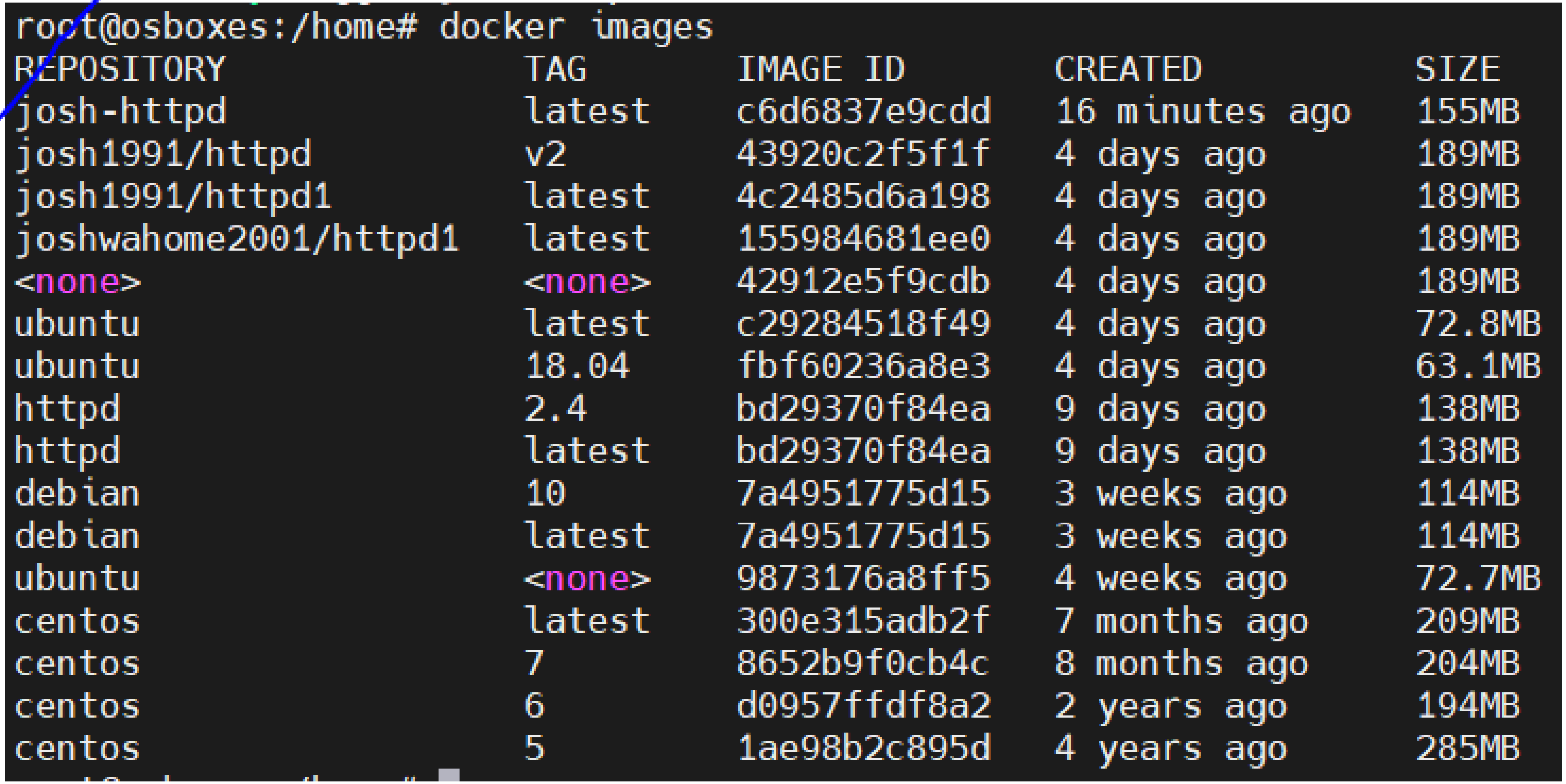
Results



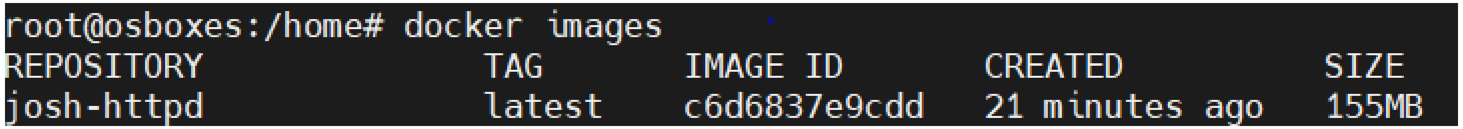
```
root@osboxes:/home# docker build -t josh-httpd .
Sending build context to Docker daemon  38.89MB
Step 1/4 : FROM httpd:2.4
 ---> bd29370f84ea
Step 2/4 : MAiNTAINER josh.kidfileapp@gmail.com
 ---> Using cache
 ---> 6e2acea31ecd
Step 3/4 : RUN apt-get update
 ---> Using cache
 ---> dad7f7f87839
Step 4/4 : COPY ./index.html /usr/local/apache2/htdocs/
 ---> Using cache
 ---> c6d6837e9cdd
Successfully built c6d6837e9cdd
Successfully tagged josh-httpd:latest
root@osboxes:/home#
```

# docker images

Results

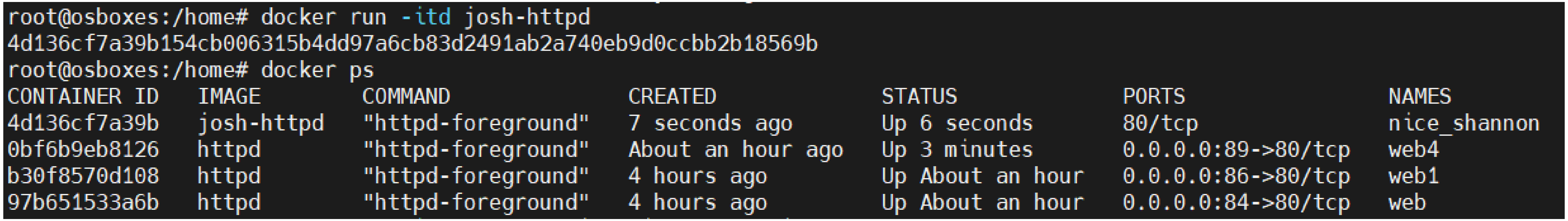


```
root@osboxes:/home# docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
josh-httpd                latest    c6d6837e9cdd   16 minutes ago   155MB
josh1991/httpd            v2        43920c2f5f1f   4 days ago       189MB
josh1991/httpd1           latest    4c2485d6a198   4 days ago       189MB
joshwahome2001/httpd1     latest    155984681ee0   4 days ago       189MB
<none>                    <none>    42912e5f9cdb   4 days ago       189MB
ubuntu                    latest    c29284518f49   4 days ago       72.8MB
ubuntu                    18.04     fbf60236a8e3   4 days ago       63.1MB
httpd                     2.4       bd29370f84ea   9 days ago       138MB
httpd                     latest    bd29370f84ea   9 days ago       138MB
debian                    10        7a4951775d15   3 weeks ago      114MB
debian                    latest    7a4951775d15   3 weeks ago      114MB
ubuntu                    <none>    9873176a8ff5   4 weeks ago      72.7MB
centos                    latest    300e315adb2f   7 months ago     209MB
centos                    7         8652b9f0cb4c   8 months ago     204MB
centos                    6         d0957ffdf8a2   2 years ago      194MB
centos                    5         1ae98b2c895d   4 years ago      285MB
```



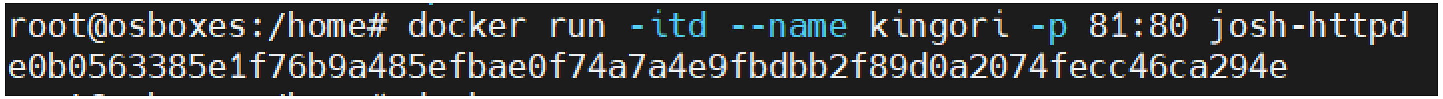
```
root@osboxes:/home# docker images
REPOSITORY                TAG        IMAGE ID        CREATED            SIZE
josh-httpd                latest     c6d6837e9cdd    21 minutes ago     155MB
```

Run the created image on docker engine as a container



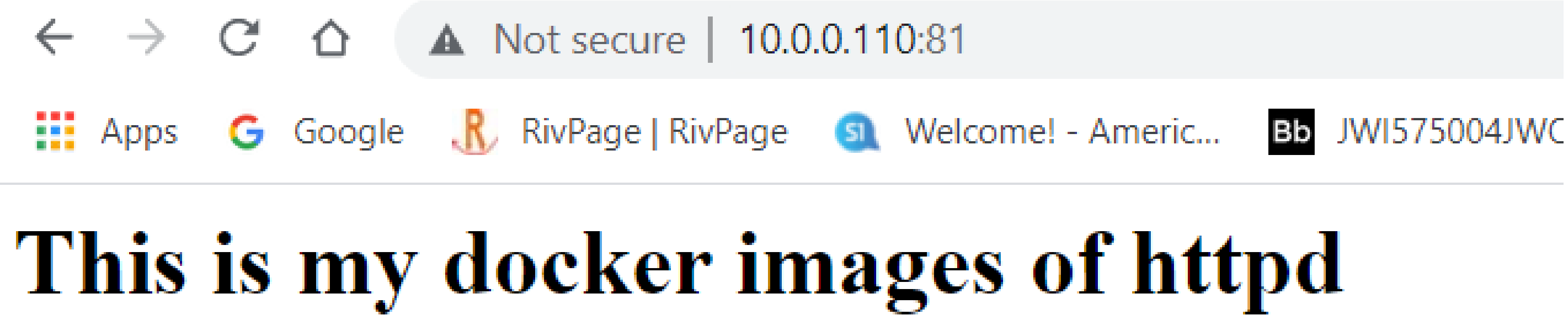
```
root@osboxes:/home# docker run -itd josh-httpd
4d136cf7a39b154cb006315b4dd97a6cb83d2491ab2a740eb9d0ccbb2b18569b
root@osboxes:/home# docker ps
CONTAINER ID   IMAGE        COMMAND              CREATED             STATUS             PORTS                  NAMES
4d136cf7a39b   josh-httpd   "httpd-foreground"   7 seconds ago       Up 6 seconds       80/tcp                 nice_shannon
0bf6b9eb8126   httpd        "httpd-foreground"   About an hour ago   Up 3 minutes       0.0.0.0:89->80/tcp     web4
b30f8570d108   httpd        "httpd-foreground"   4 hours ago         Up About an hour   0.0.0.0:86->80/tcp     web1
97b651533a6b   httpd        "httpd-foreground"   4 hours ago         Up About an hour   0.0.0.0:84->80/tcp     web
```

Add the port number and name and access the container on webserver



```
root@osboxes:/home# docker run -itd --name kingori -p 81:80 josh-httpd
e0b0563385e1f76b9a485efbae0f74a7a4e9fbdbb2f89d0a2074fecc46ca294e
```

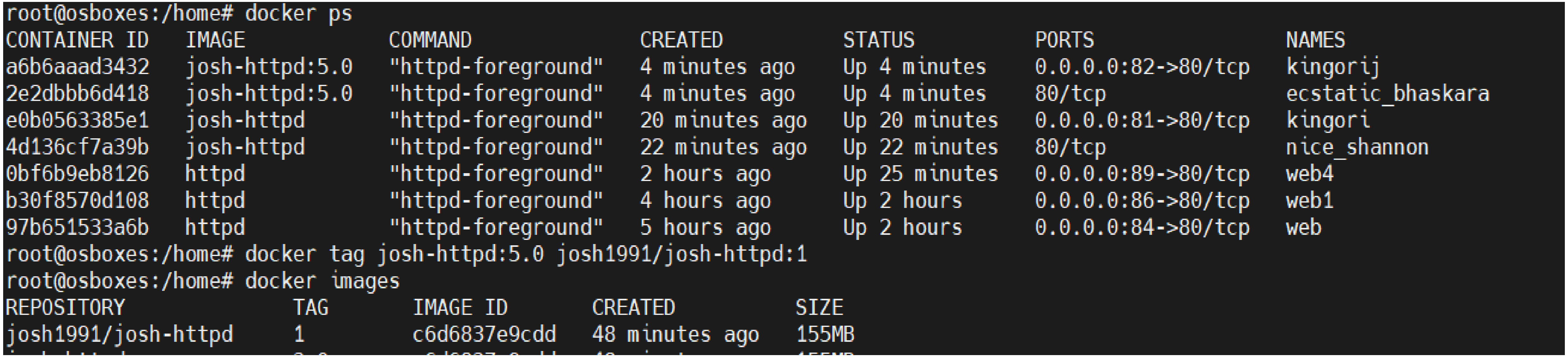
IP Address: 10.0.0.110:81



This is my docker images of httpd

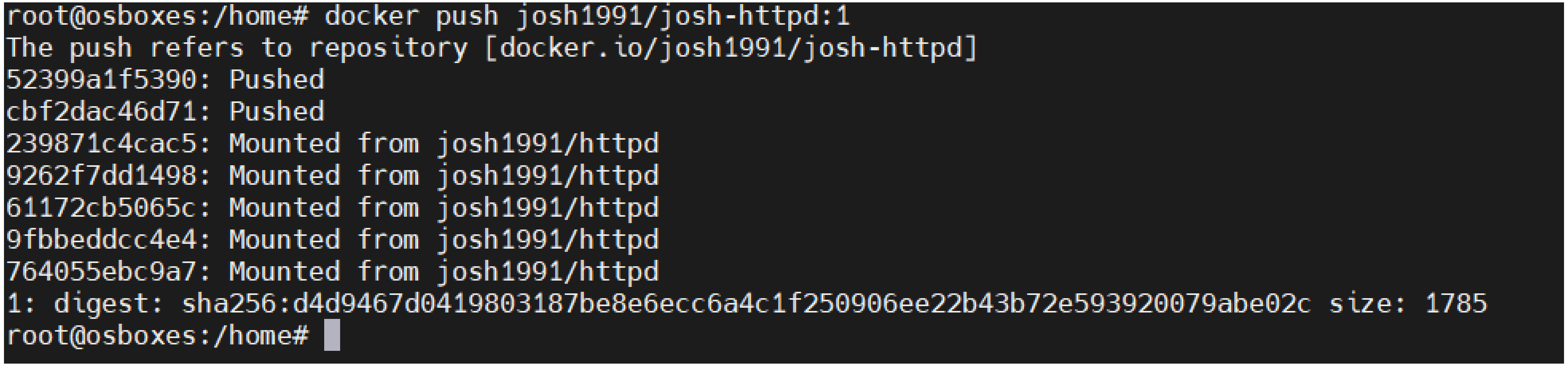
Creating a repository

# docker tag name of the image hub.docker.com account/name of the image



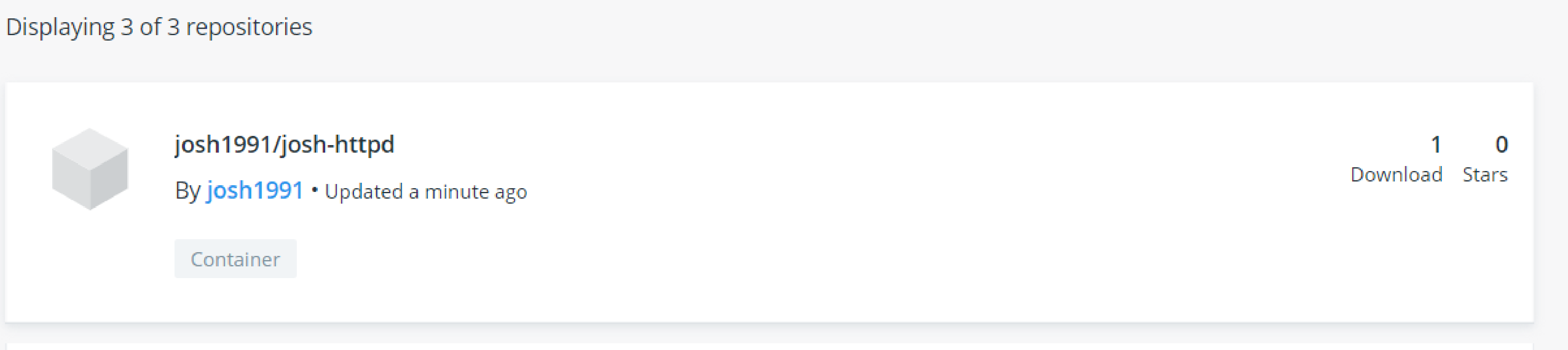
```
root@osboxes:/home# docker ps
CONTAINER ID   IMAGE            COMMAND                 CREATED          STATUS          PORTS                  NAMES
a6b6aaad3432   josh-httpd:5.0   "httpd-foreground"      4 minutes ago    Up 4 minutes    0.0.0.0:82->80/tcp     kingorij
2e2dbbb6d418   josh-httpd:5.0   "httpd-foreground"      4 minutes ago    Up 4 minutes    80/tcp                 ecstatic_bhaskara
e0b0563385e1   josh-httpd       "httpd-foreground"      20 minutes ago   Up 20 minutes   0.0.0.0:81->80/tcp     kingori
4d136cf7a39b   josh-httpd       "httpd-foreground"      22 minutes ago   Up 22 minutes   80/tcp                 nice_shannon
0bf6b9eb8126   httpd            "httpd-foreground"      2 hours ago      Up 25 minutes   0.0.0.0:89->80/tcp     web4
b30f8570d108   httpd            "httpd-foreground"      4 hours ago      Up 2 hours      0.0.0.0:86->80/tcp     web1
97b651533a6b   httpd            "httpd-foreground"      5 hours ago      Up 2 hours      0.0.0.0:84->80/tcp     web
root@osboxes:/home# docker tag josh-httpd:5.0 josh1991/josh-httpd:1
root@osboxes:/home# docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
josh1991/josh-httpd   1         c6d6837e9cdd   48 minutes ago   155MB
```

# docker push repository name:tag



```
root@osboxes:/home# docker push josh1991/josh-httpd:1
The push refers to repository [docker.io/josh1991/josh-httpd]
52399a1f5390: Pushed
cbf2dac46d71: Pushed
239871c4cac5: Mounted from josh1991/httpd
9262f7dd1498: Mounted from josh1991/httpd
61172cb5065c: Mounted from josh1991/httpd
9fbbeddcc4e4: Mounted from josh1991/httpd
764055ebc9a7: Mounted from josh1991/httpd
1: digest: sha256:d4d9467d0419803187be8e6ecc6a4c1f250906ee22b43b72e593920079abe02c size: 1785
root@osboxes:/home# 
```

Go to hub.docker.com



Displaying 3 of 3 repositories

josh1991/josh-httpd

By josh1991 • Updated a minute ago

Container

1 Download

0 Stars