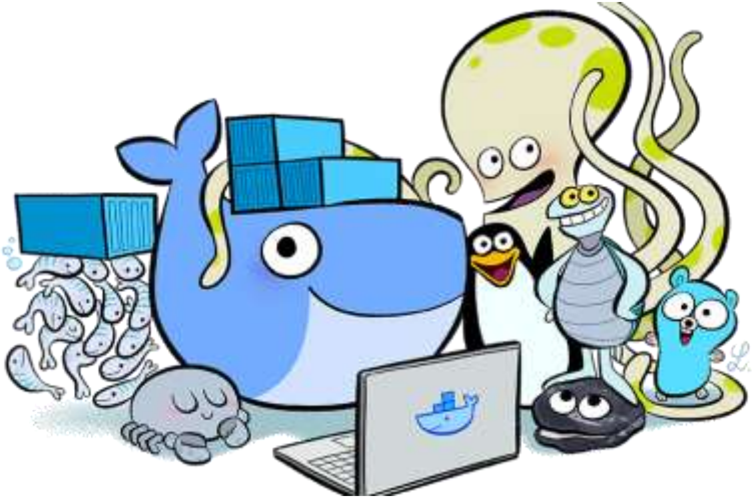
Reza Bojnordi

rezabojnordi2012@gmail.com

https://github.com/rezabojnordi

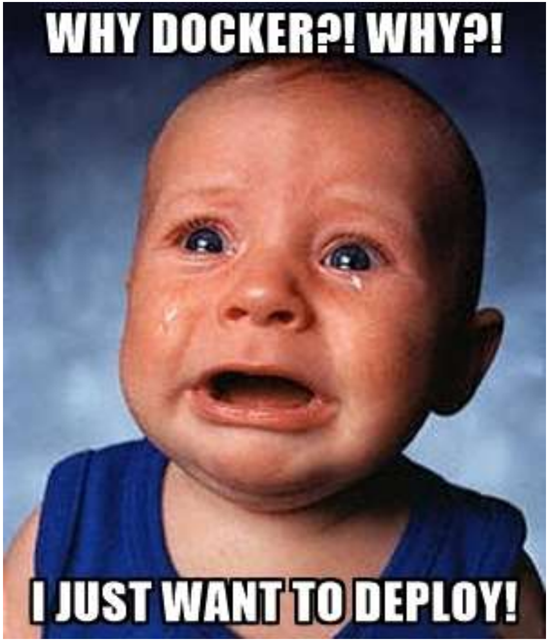
Graduated in Artificial Intelligence

t.me/cloudtalks

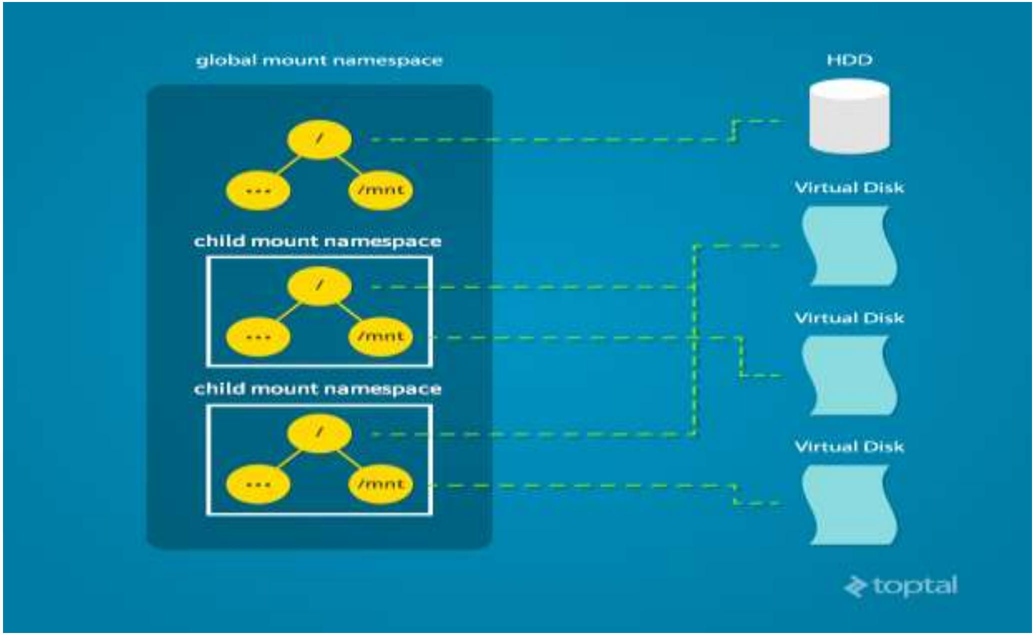


ایرانسرور

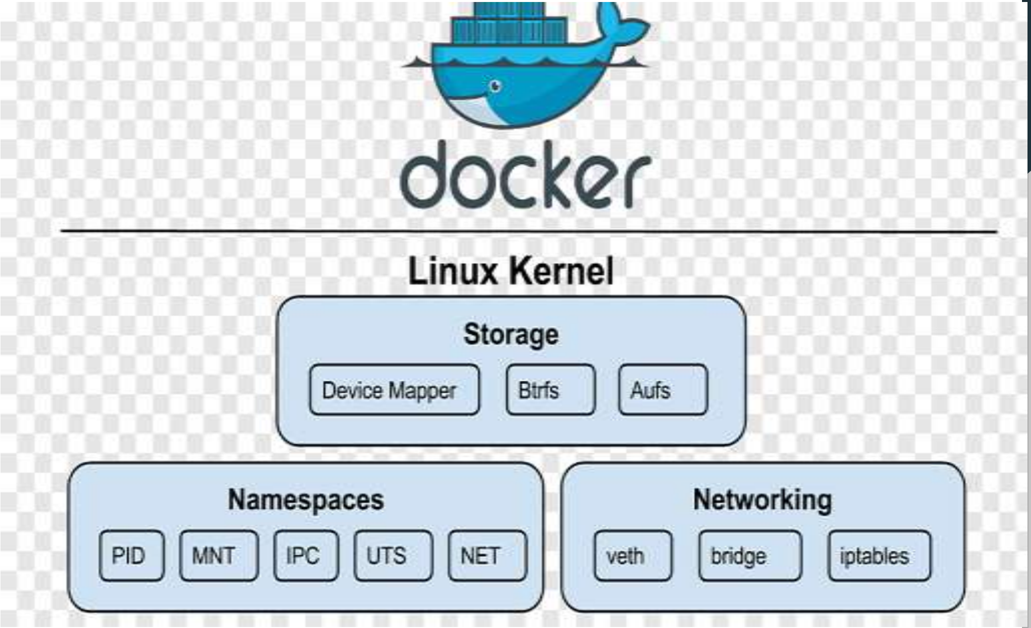
گسیناوب


WHY DOCKER?! WHY?!
I JUST WANT TO DEPLOY!

namespace

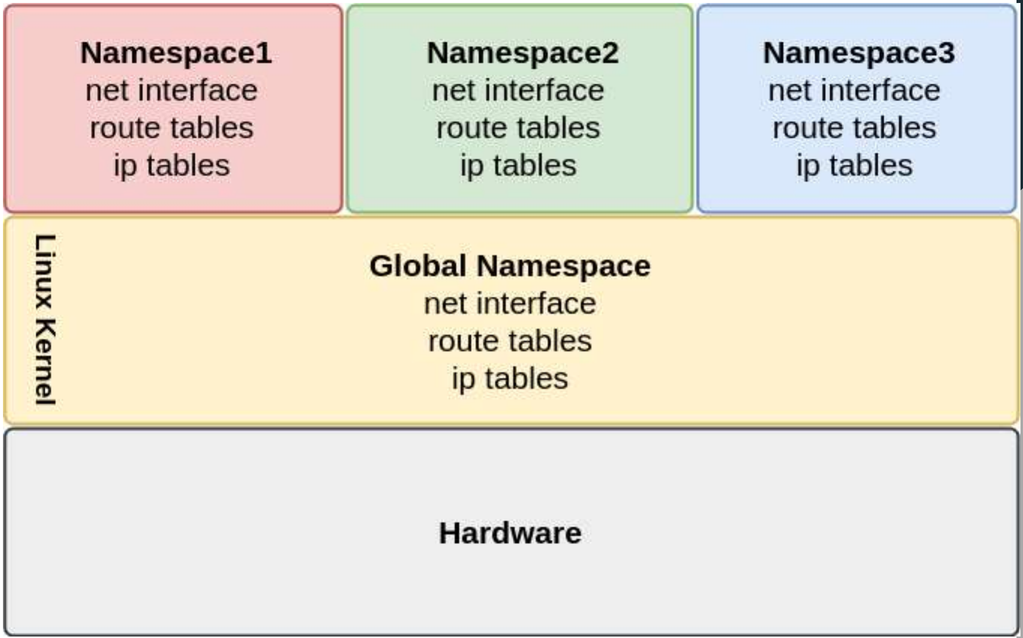

global mount namespace
/
...
/mnt
child mount namespace
/
...
/mnt
child mount namespace
/
...
/mnt
HDD
Virtual Disk
Virtual Disk
Virtual Disk
toptal

namespace



Namespace is to wrap a particular global system resource in an abstraction that makes it appear to the process within the namespace that they have their own isolated instance of the global resource. Currently there are 6 namespaces implemented in the Linux Kernel: Mount, UTS, IPC, Network, PID and User.

namespace



Namespace1
net interface
route tables
ip tables

Namespace2
net interface
route tables
ip tables

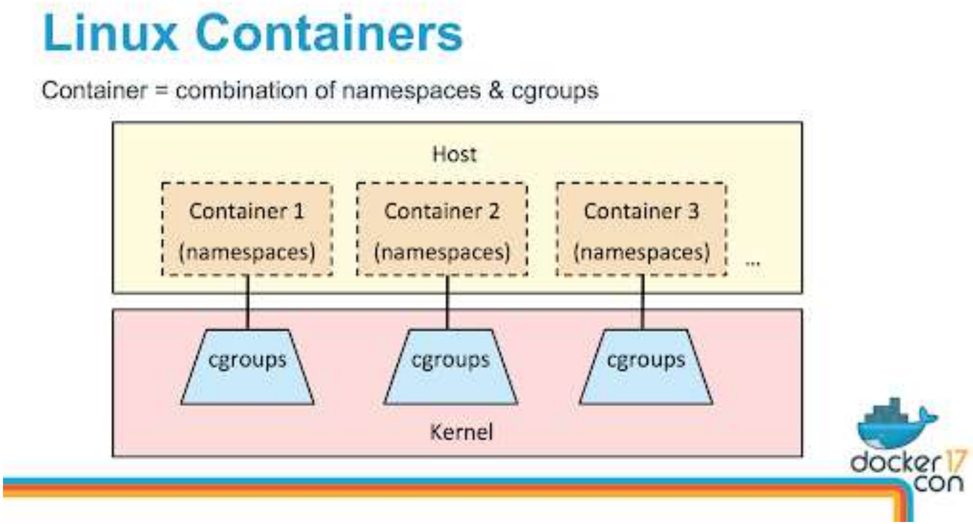
Namespace3
net interface
route tables
ip tables

Linux Kernel

Global Namespace
net interface
route tables
ip tables

Hardware

cgroup

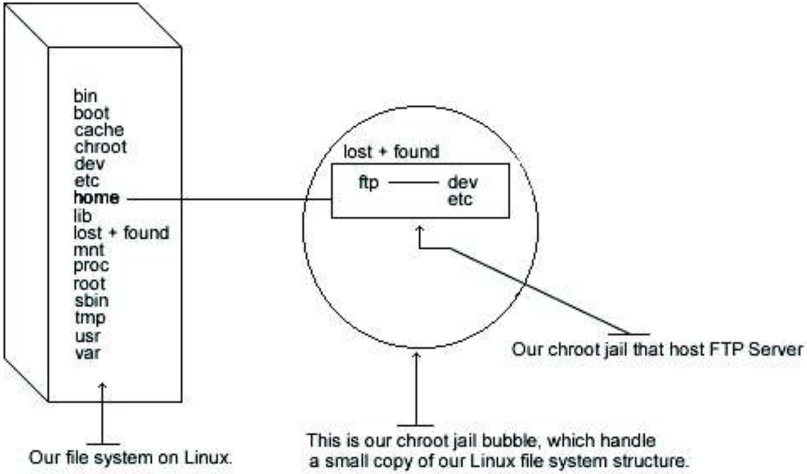


Linux Containers

Container = combination of namespaces & cgroups

Cgroups allow allocating resources to user-defined groups of processes running on the system. A cgroup limits an application to a specific set of resources.

chroot



bin
boot
cache
chroot
dev
etc
home
lib
lost + found
mnt
proc
root
sbin
tmp
usr
var

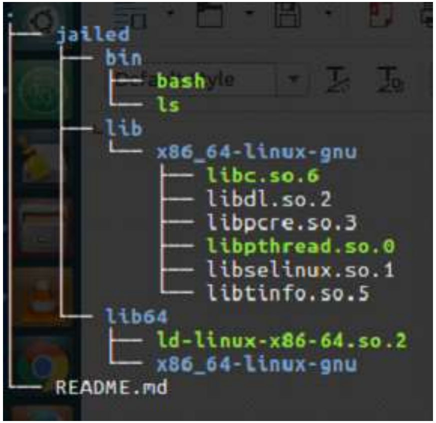
lost + found

ftp — dev
etc

Our chroot jail that host FTP Server

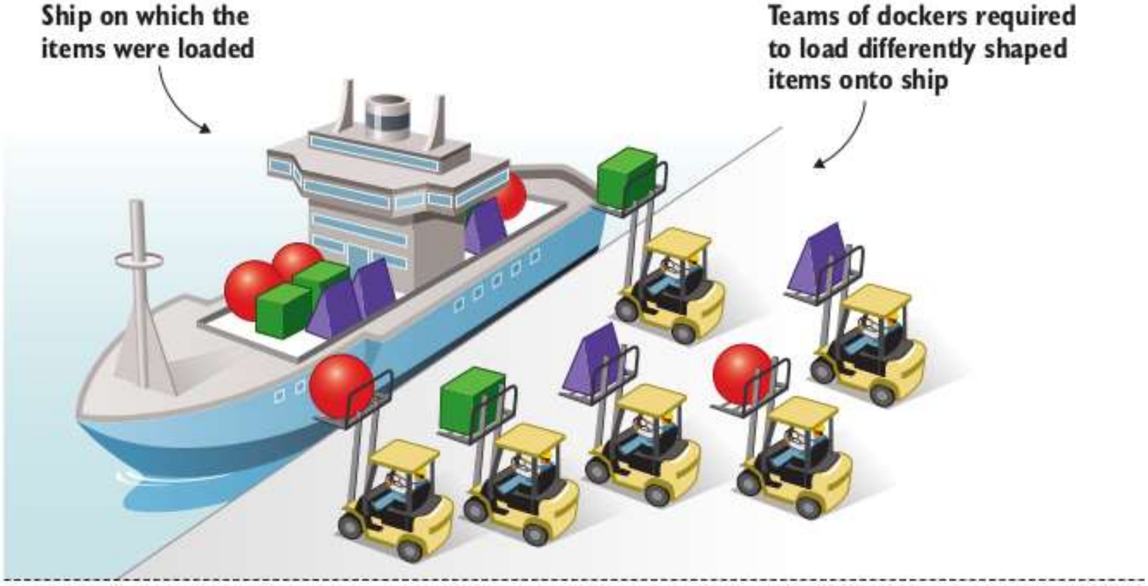
Our file system on Linux.

This is our chroot jail bubble, which handle
a small copy of our Linux file system structure.



```
├── jailed
│   ├── bin
│   │   ├── bash
│   │   └── ls
│   ├── lib
│   │   └── x86_64-linux-gnu
│   │       ├── libc.so.6
│   │       ├── libdl.so.2
│   │       ├── libpcre.so.3
│   │       ├── libpthread.so.0
│   │       ├── libselinux.so.1
│   │       └── libtinfo.so.5
│   └── lib64
│       ├── ld-linux-x86-64.so.2
│       └── x86_64-linux-gnu
└── README.md
```

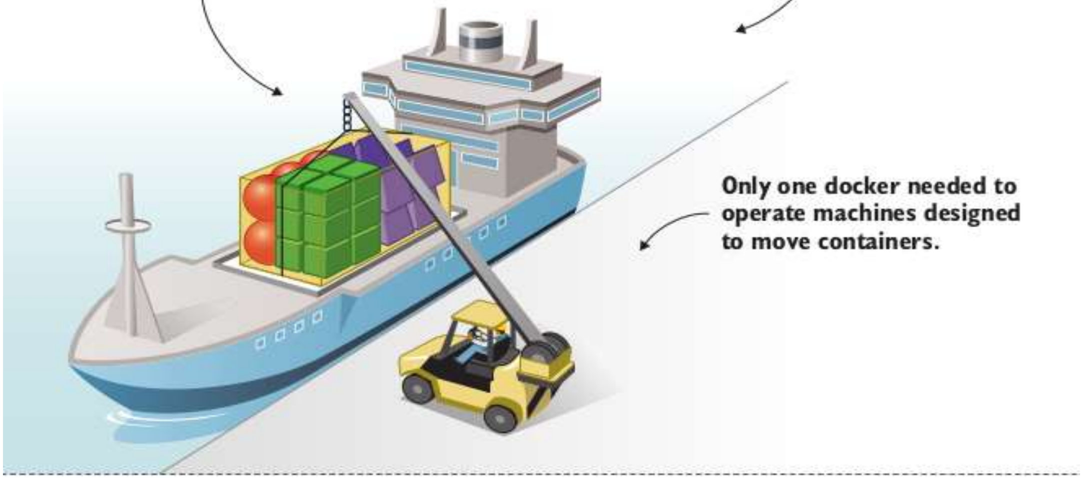
What is docker



Ship on which the items were loaded

Teams of dockers required to load differently shaped items onto ship

What is docker



Single container with different items in it. It doesn't matter to the carrier what's inside the container. The carrier can be loaded up elsewhere, reducing the bottleneck of loading at port.

Ship can be designed to carry, load, and unload predictably shaped items more efficiently.

Only one docker needed to operate machines designed to move containers.

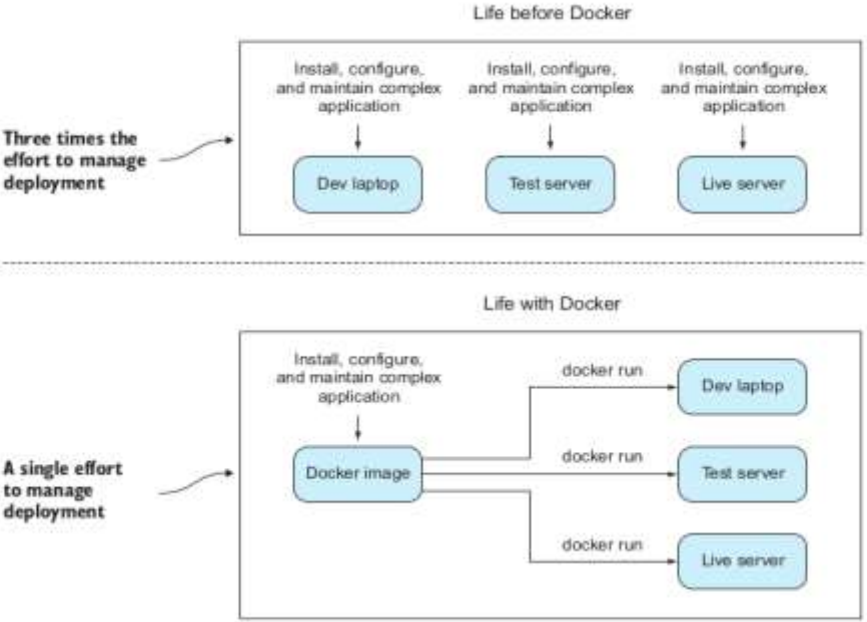
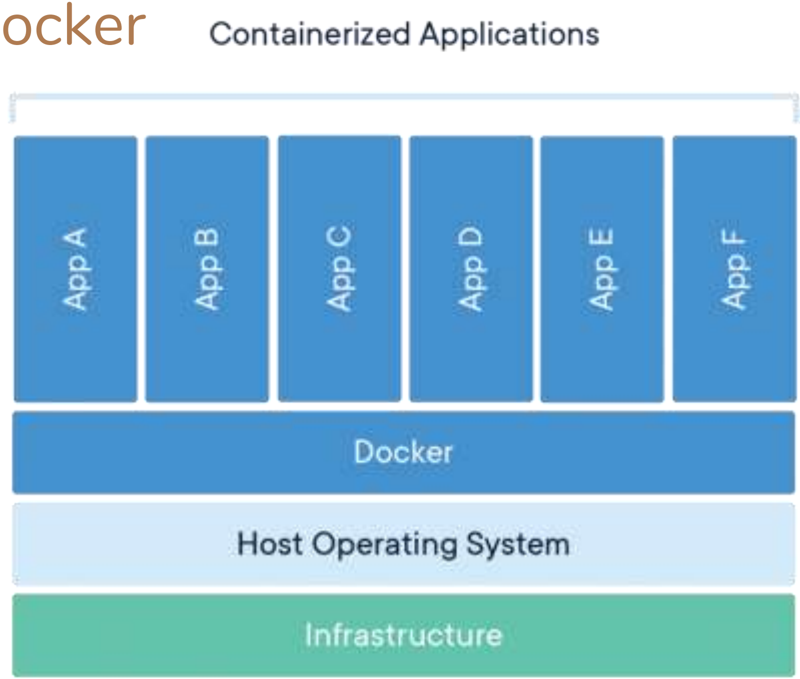
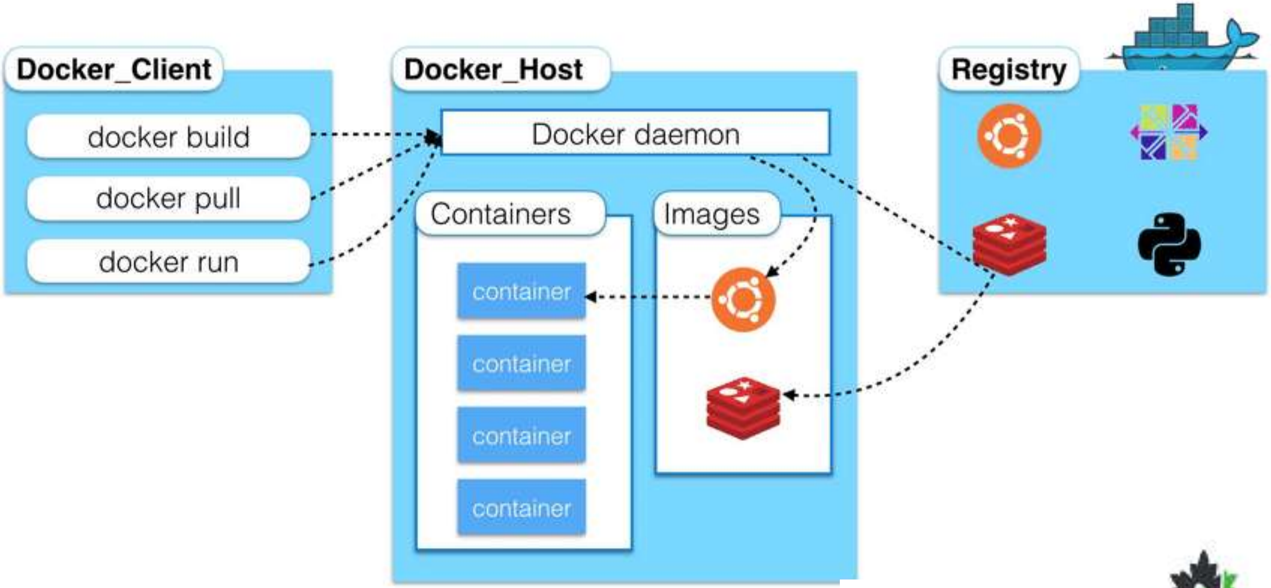

Life before Docker
Three times the effort to manage deployment
Install, configure, and maintain complex application
Install, configure, and maintain complex application
Install, configure, and maintain complex application
Dev laptop
Test server
Live server
Life with Docker
A single effort to manage deployment
Install, configure, and maintain complex application
Docker image
docker run
docker run
docker run
Dev laptop
Test server
Live server

Figure 1.3 Software delivery before and after Docker

Container in docker



Docker and Register



Vm vs Docker

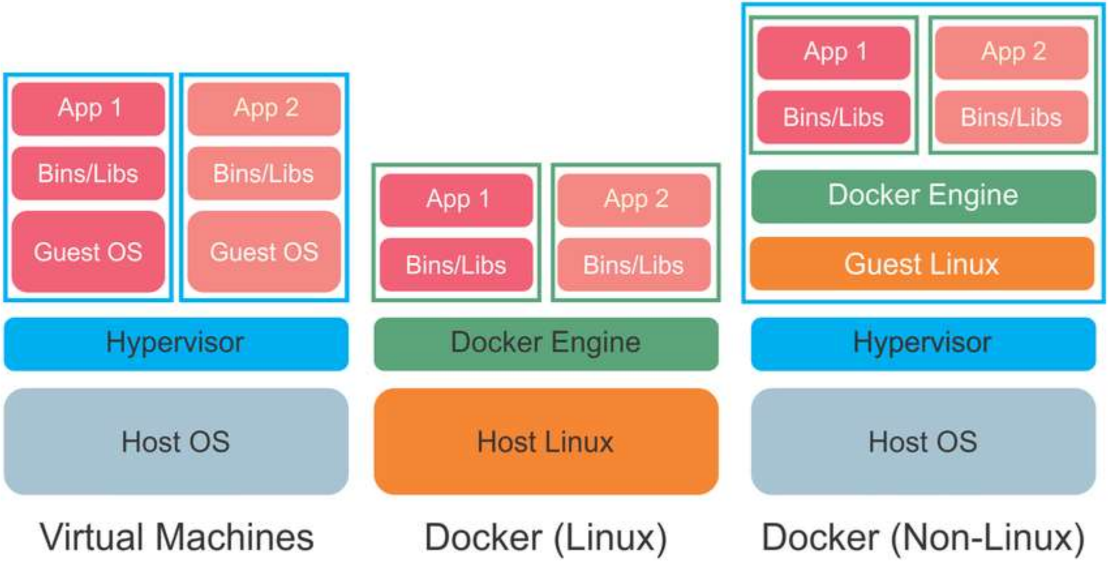

App 1
App 2
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Hypervisor
Host OS
Virtual Machines
App 1
App 2
Bins/Libs
Bins/Libs
Docker Engine
Host Linux
Docker (Linux)
App 1
App 2
Bins/Libs
Bins/Libs
Docker Engine
Guest Linux
Hypervisor
Host OS
Docker (Non-Linux)

Image in Docker

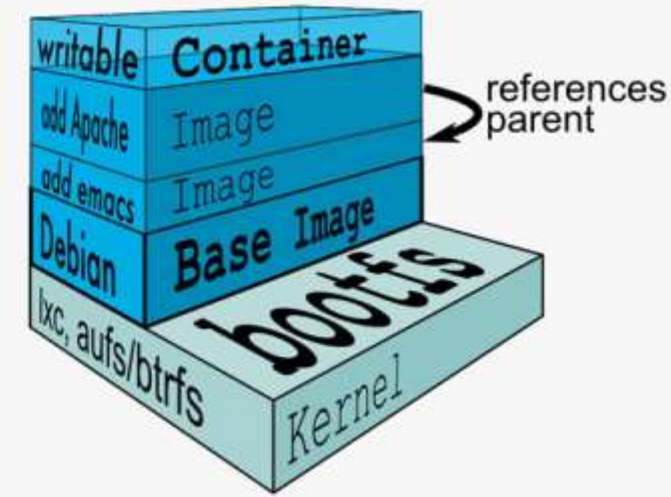


Image in Docker

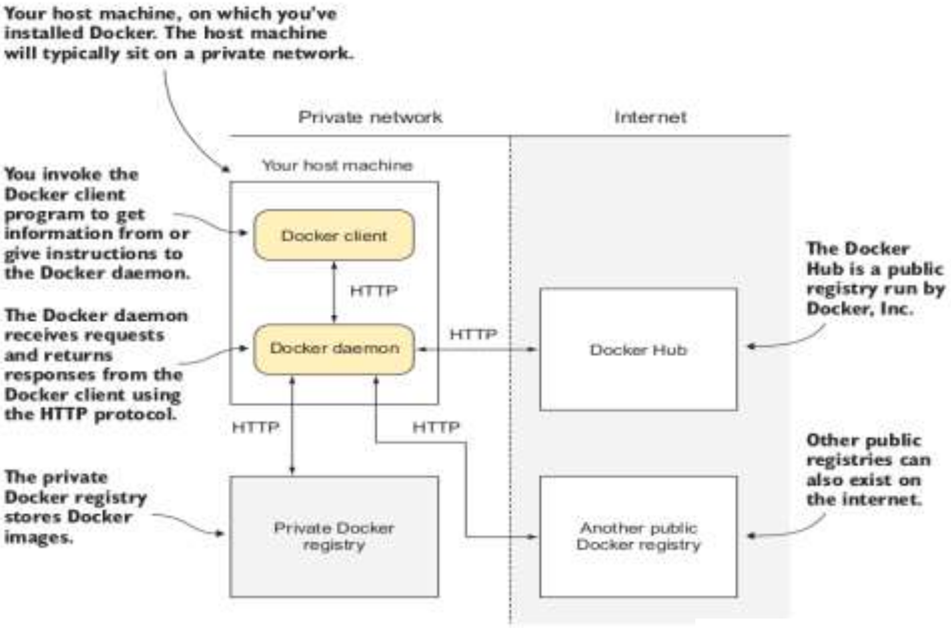
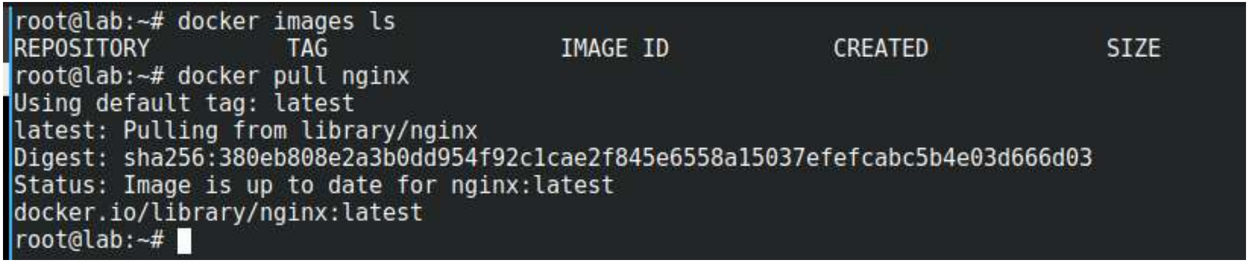

Your host machine, on which you've installed Docker. The host machine will typically sit on a private network.
Private network
Internet
Your host machine
You invoke the Docker client program to get information from or give instructions to the Docker daemon.
Docker client
HTTP
The Docker daemon receives requests and returns responses from the Docker client using the HTTP protocol.
Docker daemon
HTTP
Docker Hub
The Docker Hub is a public registry run by Docker, Inc.
HTTP
HTTP
Other public registries can also exist on the internet.
The private Docker registry stores Docker images.
Private Docker registry
Another public Docker registry

Image in Docker

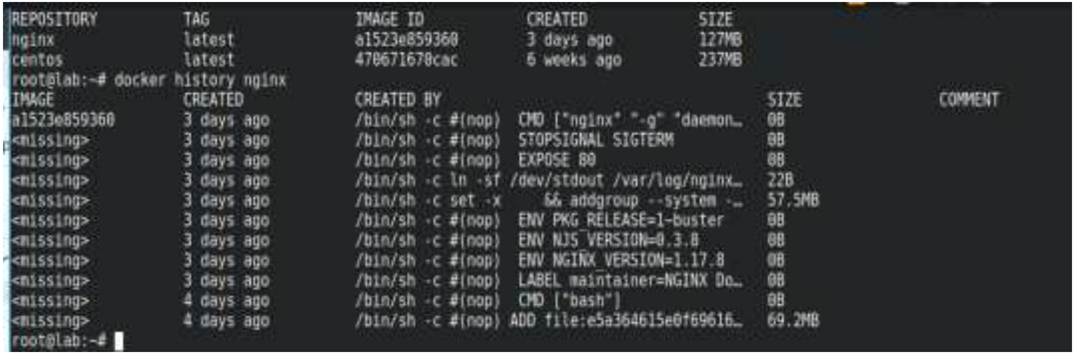
- docker image ls
- docker pull nginx



```
root@lab:~# docker images ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
root@lab:~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:380eb808e2a3b0dd954f92c1cae2f845e6558a15037efefcabc5b4e03d666d03
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
root@lab:~# 
```

Image in Docker

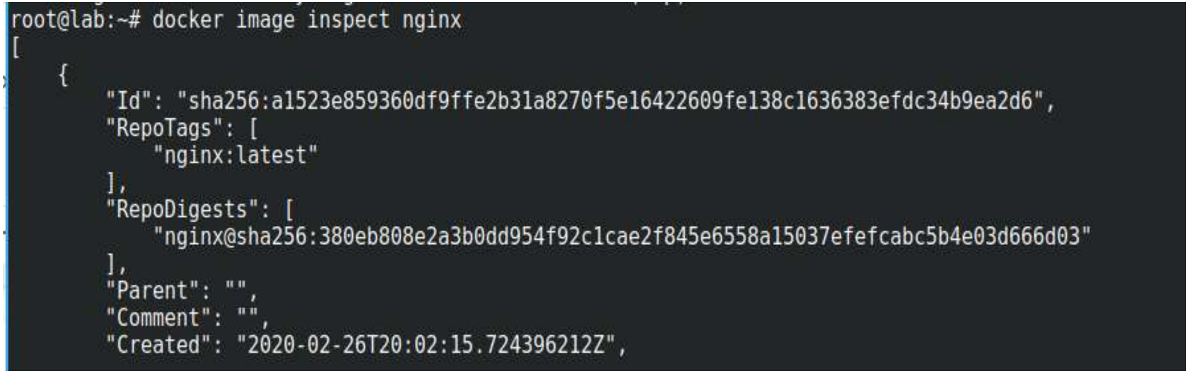
- docker history nginx



```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              a1523e859360        3 days ago          127MB
centos              latest              470671670cac        6 weeks ago         237MB
root@lab:~# docker history nginx
IMAGE               CREATED             CREATED BY                                      SIZE                COMMENT
a1523e859360        3 days ago          /bin/sh -c #(nop)  CMD ["nginx" "-g" "daemon…   0B
<missing>           3 days ago          /bin/sh -c #(nop)  STOPSIGNAL SIGTERM           0B
<missing>           3 days ago          /bin/sh -c #(nop)  EXPOSE 80                    0B
<missing>           3 days ago          /bin/sh -c ln -sf /dev/stdout /var/log/nginx…   22B
<missing>           3 days ago          /bin/sh -c set -x     && addgroup --system -…   57.5MB
<missing>           3 days ago          /bin/sh -c #(nop)  ENV PKG_RELEASE=1~buster     0B
<missing>           3 days ago          /bin/sh -c #(nop)  ENV NJS_VERSION=0.3.8        0B
<missing>           3 days ago          /bin/sh -c #(nop)  ENV NGINX_VERSION=1.17.8     0B
<missing>           3 days ago          /bin/sh -c #(nop)  LABEL maintainer=NGINX Do…   0B
<missing>           4 days ago          /bin/sh -c #(nop)  CMD ["bash"]                 0B
<missing>           4 days ago          /bin/sh -c #(nop) ADD file:e5a364615e0f69616…   69.2MB
root@lab:~# █
```

Image in Docker

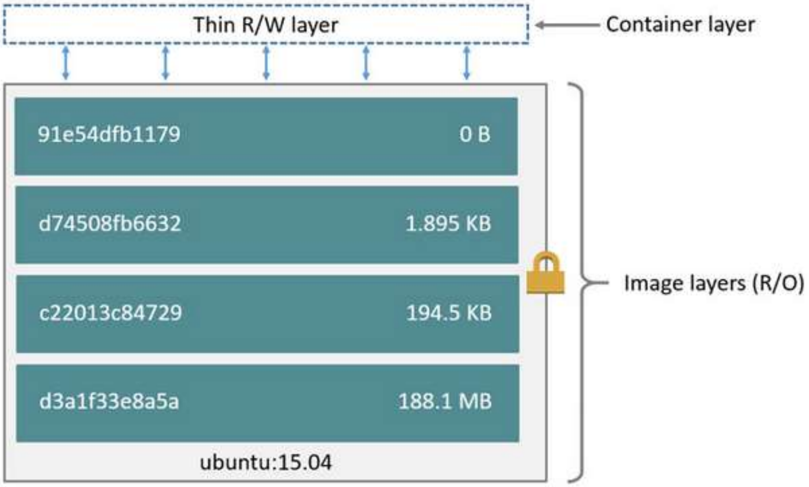
- docker inspect nginx



```
root@lab:~# docker image inspect nginx
[
    {
        "Id": "sha256:a1523e859360df9ffe2b31a8270f5e16422609fe138c1636383efdc34b9ea2d6",
        "RepoTags": [
            "nginx:latest"
        ],
        "RepoDigests": [
            "nginx@sha256:380eb808e2a3b0dd954f92c1cae2f845e6558a15037efefcabc5b4e03d666d03"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2020-02-26T20:02:15.724396212Z",
```

Images and layers

```
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```



Thin R/W layer

Container layer

91e54dfb1179 0 B

d74508fb6632 1.895 KB

c22013c84729 194.5 KB

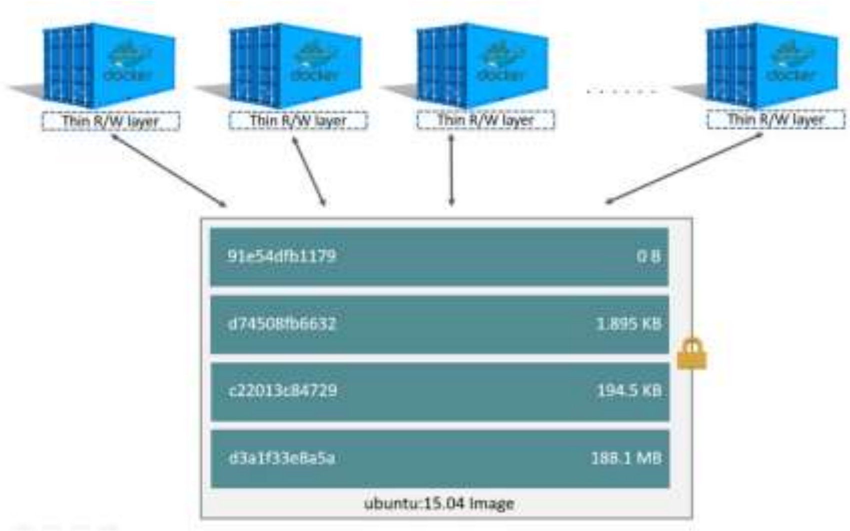
d3a1f33e8a5a 188.1 MB

ubuntu:15.04

Image layers (R/O)

Container
(based on ubuntu:15.04 image)

Images and layers



Thin R/W layer

Thin R/W layer

Thin R/W layer

Thin R/W layer

91e54dfb1179 0 B

d74508fb6632 1.895 KB

c22013c84729 194.5 KB

d3a1f33e8a5a 188.1 MB

ubuntu:15.04 image

Images and layers

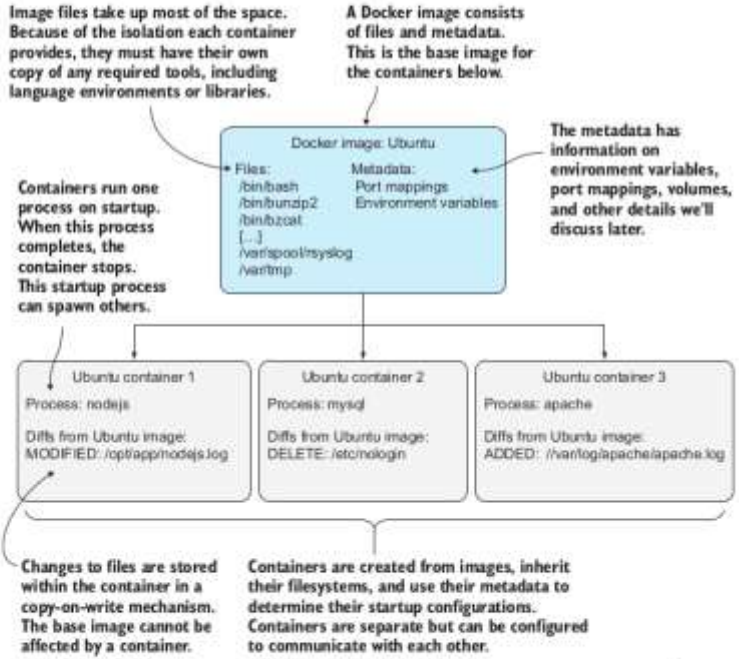


Image files take up most of the space. Because of the isolation each container provides, they must have their own copy of any required tools, including language environments or libraries.

A Docker image consists of files and metadata. This is the base image for the containers below.

The metadata has information on environment variables, port mappings, volumes, and other details we'll discuss later.

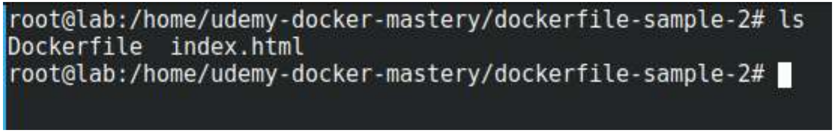
Containers run one process on startup. When this process completes, the container stops. This startup process can spawn others.

Changes to files are stored within the container in a copy-on-write mechanism. The base image cannot be affected by a container.

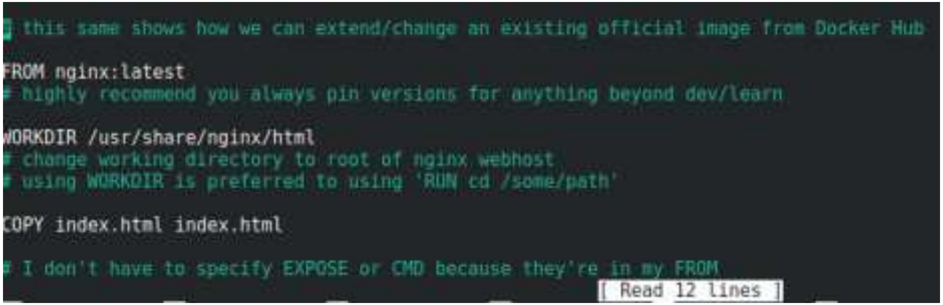
Containers are created from images, inherit their filesystems, and use their metadata to determine their startup configurations. Containers are separate but can be configured to communicate with each other.

Figure 1.5 Docker images and containers

Dockerfile



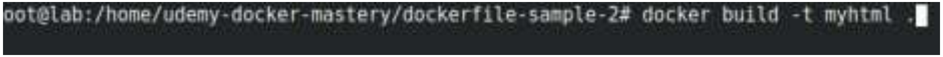
```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# ls
Dockerfile  index.html
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```



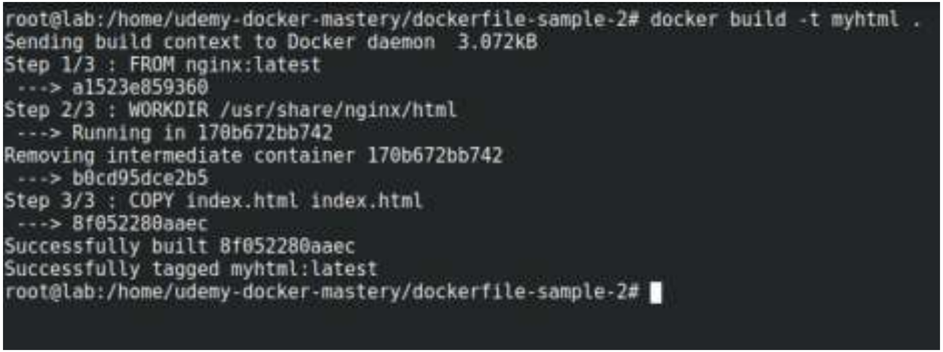
```
# this same shows how we can extend/change an existing official image from Docker Hub

FROM nginx:latest
# highly recommend you always pin versions for anything beyond dev/learn

WORKDIR /usr/share/nginx/html
# change working directory to root of nginx webhost
# using WORKDIR is preferred to using 'RUN cd /some/path'

COPY index.html index.html

# I don't have to specify EXPOSE or CMD because they're in my FROM
                                                  [ Read 12 lines ]
```

Dockerfile

- `docker build -t myhtml .`

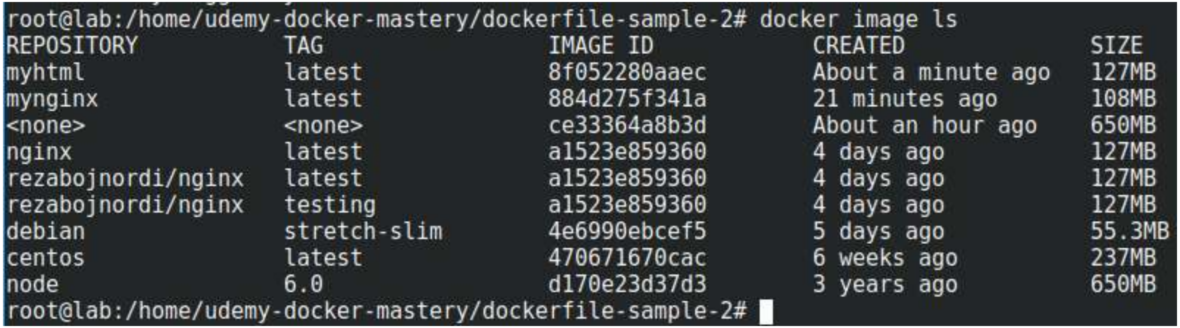


```
oot@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker build -t myhtml .
```



```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker build -t myhtml .
Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM nginx:latest
 ---> a1523e859360
Step 2/3 : WORKDIR /usr/share/nginx/html
 ---> Running in 170b672bb742
Removing intermediate container 170b672bb742
 ---> b0cd95dce2b5
Step 3/3 : COPY index.html index.html
 ---> 8f052280aaec
Successfully built 8f052280aaec
Successfully tagged myhtml:latest
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```

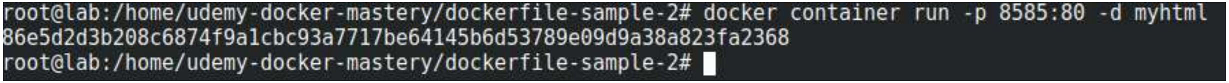
Image list



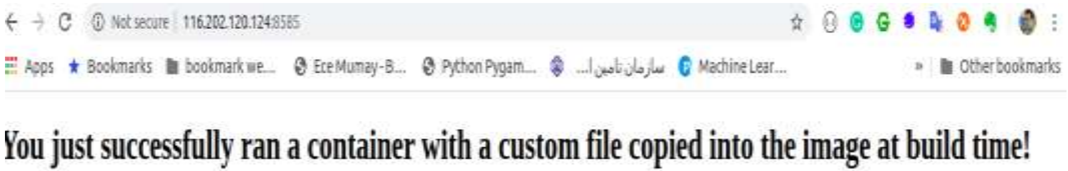
```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker image ls
REPOSITORY           TAG                 IMAGE ID            CREATED              SIZE
myhtml               latest              8f052280aaec        About a minute ago   127MB
mynginx              latest              884d275f341a        21 minutes ago       108MB
<none>               <none>              ce33364a8b3d        About an hour ago    650MB
nginx                latest              a1523e859360        4 days ago           127MB
rezabojnordi/nginx   latest              a1523e859360        4 days ago           127MB
rezabojnordi/nginx   testing             a1523e859360        4 days ago           127MB
debian               stretch-slim        4e6990ebcef5        5 days ago           55.3MB
centos               latest              470671670cac        6 weeks ago          237MB
node                 6.0                 d170e23d37d3        3 years ago          650MB
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```

container

- `docker container run -p 8585:80 -d myhtml`



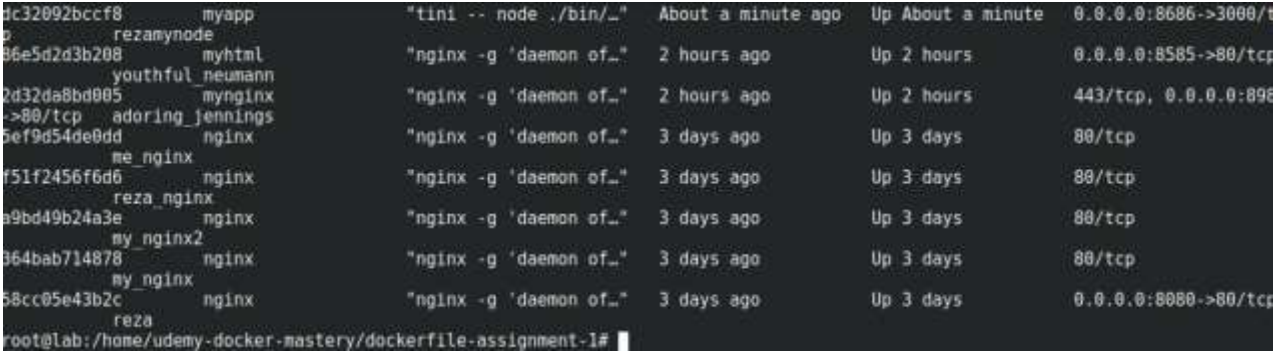
```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker container run -p 8585:80 -d myhtml
86e5d2d3b208c6874f9a1cbc93a7717be64145b6d53789e09d9a38a823fa2368
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```



You just successfully ran a container with a custom file copied into the image at build time!

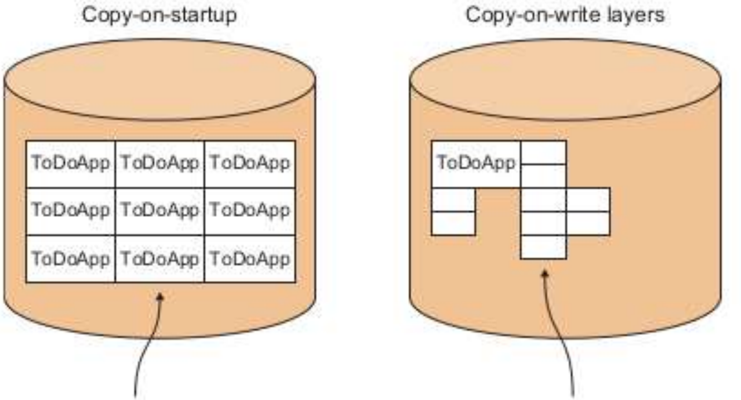
container

- `docker container ls`



```
dc32092bccf8        myapp               "tini -- node ./bin/…"   About a minute ago   Up About a minute   0.0.0.0:8686->3000/t
p           rezamynode
86e5d2d3b208        myhtml              "nginx -g 'daemon of…"   2 hours ago          Up 2 hours          0.0.0.0:8585->80/tcp
            youthful_neumann
2d32da8bd005        mynginx             "nginx -g 'daemon of…"   2 hours ago          Up 2 hours          443/tcp, 0.0.0.0:898
->80/tcp    adoring_jennings
5ef9d54de0dd        nginx               "nginx -g 'daemon of…"   3 days ago           Up 3 days           80/tcp
            me_nginx
f51f2456f6d6        nginx               "nginx -g 'daemon of…"   3 days ago           Up 3 days           80/tcp
            reza_nginx
a9bd49b24a3e        nginx               "nginx -g 'daemon of…"   3 days ago           Up 3 days           80/tcp
            my_nginx2
364bab714878        nginx               "nginx -g 'daemon of…"   3 days ago           Up 3 days           80/tcp
            my_nginx
58cc05e43b2c        nginx               "nginx -g 'daemon of…"   3 days ago           Up 3 days           0.0.0.0:8080->80/tcp
            reza
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# 
```

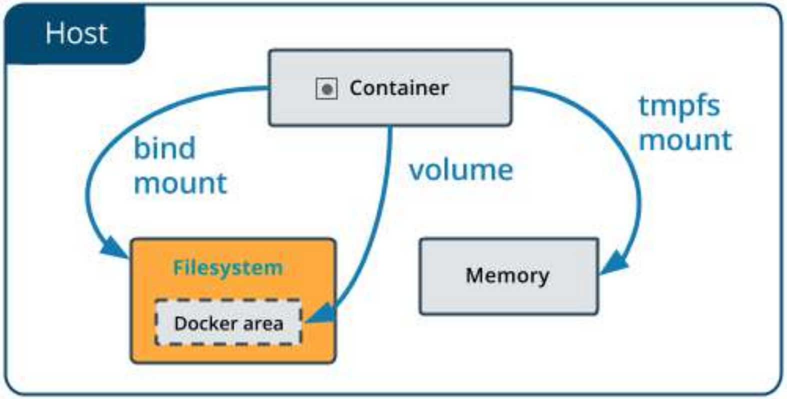
Container and Copy-on-Write



A non-layered application with nine copies made on disk for nine running instances.

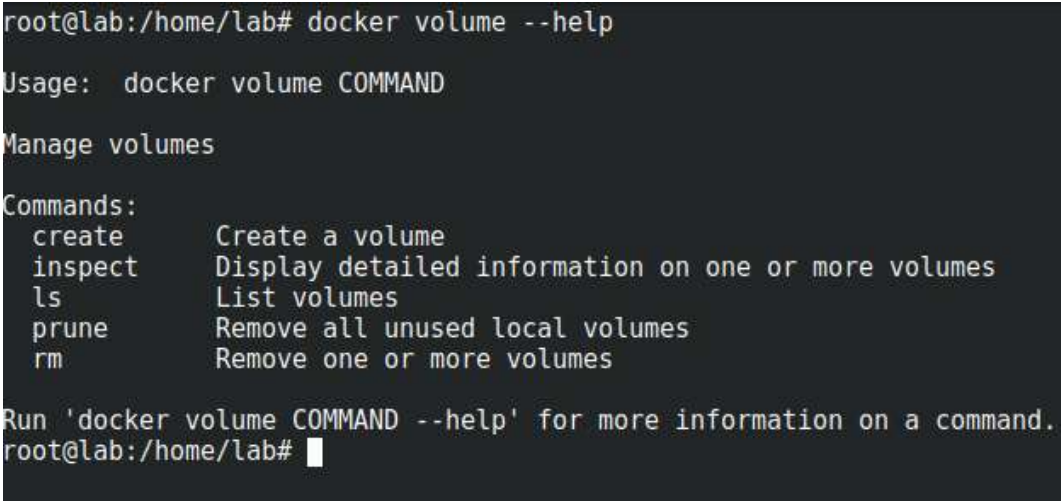
Each block represents a running container's file differences from the original ToDoApp's image. This uses much less disk space.

Docker Volume



Docker Volume

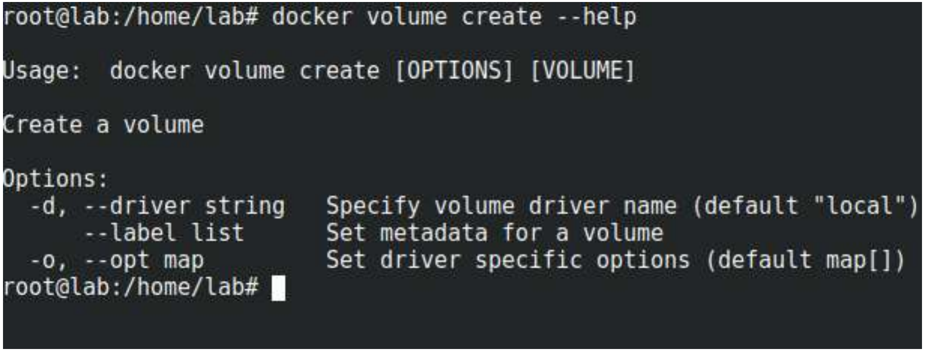
- `docker volume --help`



```
root@lab:/home/lab# docker volume --help

Usage:  docker volume COMMAND

Manage volumes

Commands:
  create      Create a volume
  inspect     Display detailed information on one or more volumes
  ls          List volumes
  prune       Remove all unused local volumes
  rm          Remove one or more volumes

Run 'docker volume COMMAND --help' for more information on a command.
root@lab:/home/lab# █
```

Docker Volume

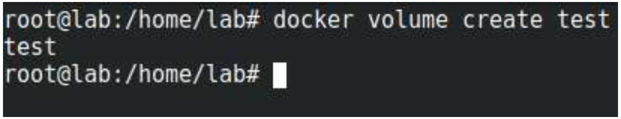
- `docker volume create --help`



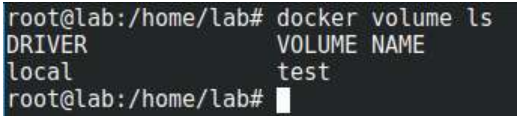
```
root@lab:/home/lab# docker volume create --help

Usage:  docker volume create [OPTIONS] [VOLUME]

Create a volume

Options:
  -d, --driver string   Specify volume driver name (default "local")
      --label list      Set metadata for a volume
  -o, --opt map         Set driver specific options (default map[])
root@lab:/home/lab# 
```

Docker Volume

- docker volume create test
- Docker volume ls



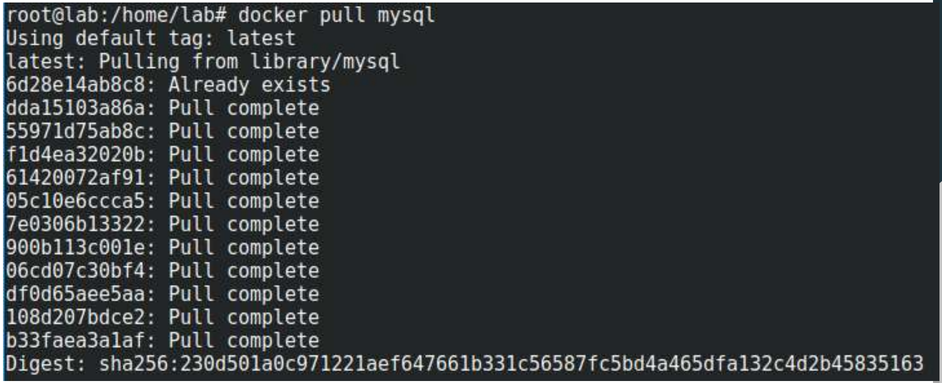
```
root@lab:/home/lab# docker volume create test
test
root@lab:/home/lab# 
```



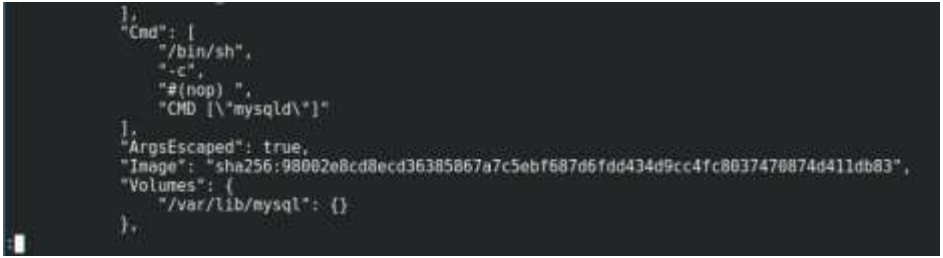
```
root@lab:/home/lab# docker volume ls
DRIVER              VOLUME NAME
local               test
root@lab:/home/lab# 
```

Docker Volume

- `docker pull mysql`
- `docker image \`
  `inspect mysql`



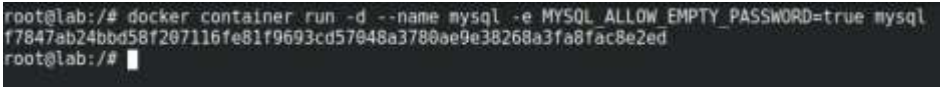
```
root@lab:/home/lab# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
6d28e14ab8c8: Already exists
dda15103a86a: Pull complete
55971d75ab8c: Pull complete
f1d4ea32020b: Pull complete
61420072af91: Pull complete
05c10e6ccca5: Pull complete
7e0306b13322: Pull complete
900b113c001e: Pull complete
06cd07c30bf4: Pull complete
df0d65aee5aa: Pull complete
108d207bdce2: Pull complete
b33faea3a1af: Pull complete
Digest: sha256:230d501a0c971221aef647661b331c56587fc5bd4a465dfa132c4d2b45835163
```



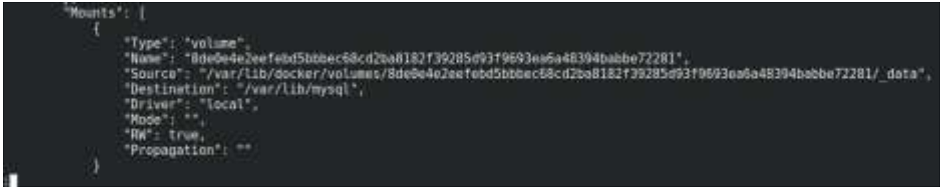
```
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"mysqld\"]"
            ],
            "ArgsEscaped": true,
            "Image": "sha256:98002e8cd8ecd36385867a7c5ebf687d6fdd434d9cc4fc8037470874d411db83",
            "Volumes": {
                "/var/lib/mysql": {}
            },
:
```

Docker Volume

- `docker container run -d --name mysql -e \`
  `MYSQL_ALLOW_EMPTY_PASSWORD=true mysql`
- `docker container inspect mysql`



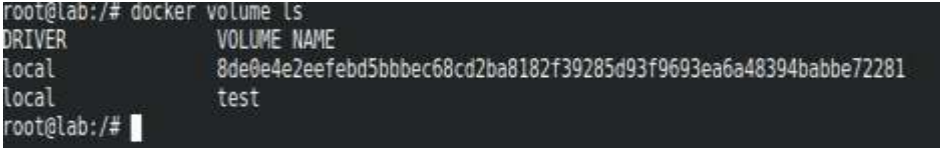
```
root@lab:/# docker container run -d --name mysql -e MYSQL_ALLOW_EMPTY_PASSWORD=true mysql
f7847ab24bbd58f207116fe81f9693cd57048a3780ae9e38268a3fa8fac8e2ed
root@lab:/# 
```



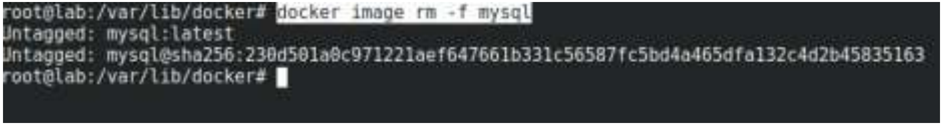
```
        "Mounts": [
            {
                "Type": "volume",
                "Name": "8de0e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281",
                "Source": "/var/lib/docker/volumes/8de0e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281/_data",
                "Destination": "/var/lib/mysql",
                "Driver": "local",
                "Mode": "",
                "RW": true,
                "Propagation": ""
            }
```

Docker Volume

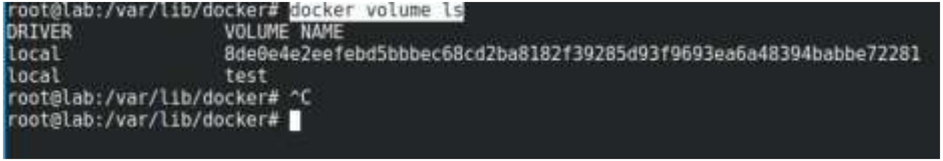
- docker volume ls
- `docker image rm -f mysql`
- `docker volume ls`



```
root@lab:/# docker volume ls
DRIVER              VOLUME NAME
local               8de0e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281
local               test
root@lab:/#
```



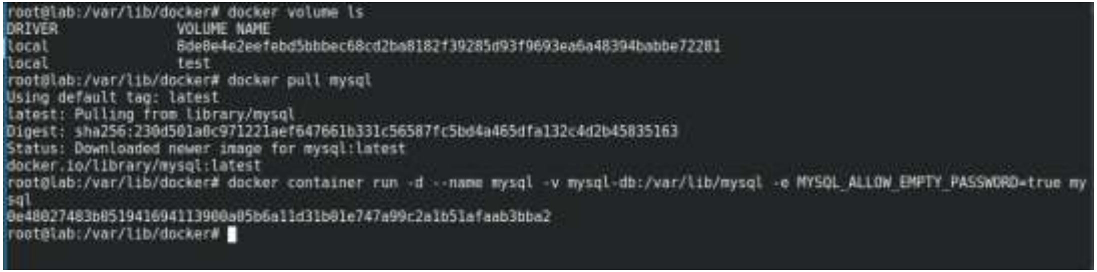
```
root@lab:/var/lib/docker# docker image rm -f mysql
Untagged: mysql:latest
Untagged: mysql@sha256:230d501a0c971221aef647661b331c56587fc5bd4a465dfa132c4d2b45835163
root@lab:/var/lib/docker#
```



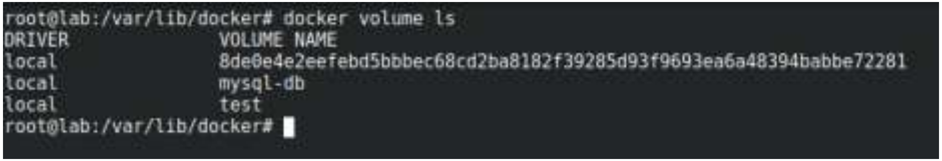
```
root@lab:/var/lib/docker# docker volume ls
DRIVER              VOLUME NAME
local               8de0e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281
local               test
root@lab:/var/lib/docker# ^C
root@lab:/var/lib/docker#
```

Docker Volume

- `docker container run -d --name mysql -v mysql-db:/var/lib/mysql -e MYSQL_ALLOW_EMPTY_PASSWORD=true mysql`

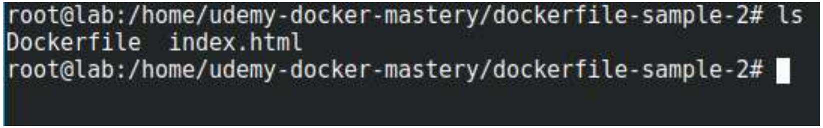


```
root@lab:/var/lib/docker# docker volume ls
DRIVER              VOLUME NAME
local               8de8e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281
local               test
root@lab:/var/lib/docker# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
Digest: sha256:230d501a0c971221aef647661b331c56587fc5bd4a465dfa132c4d2b45835163
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
root@lab:/var/lib/docker# docker container run -d --name mysql -v mysql-db:/var/lib/mysql -e MYSQL_ALLOW_EMPTY_PASSWORD=true my
sql
0e48027483b0519416941139000a05b6a11d31b01e747a99c2a1b51afaab3bba2
root@lab:/var/lib/docker# █
```



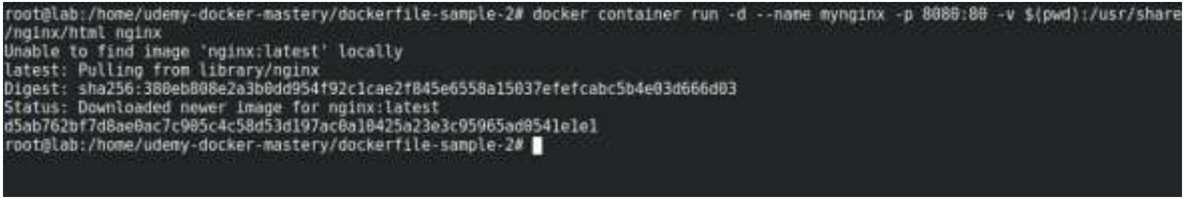
```
root@lab:/var/lib/docker# docker volume ls
DRIVER              VOLUME NAME
local               8de0e4e2eefebd5bbbec68cd2ba8182f39285d93f9693ea6a48394babbe72281
local               mysql-db
local               test
root@lab:/var/lib/docker# █
```

Bind mount

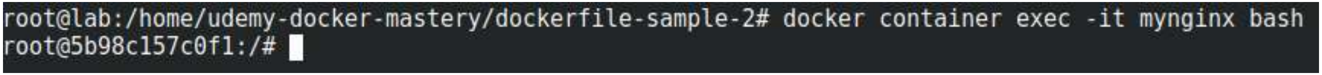


```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# ls
Dockerfile  index.html
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```

- docker container run -d --name mynginx -p 8080:80 -v $(pwd):/usr/share/nginx/html nginx
- docker container exec -it mynginx bash

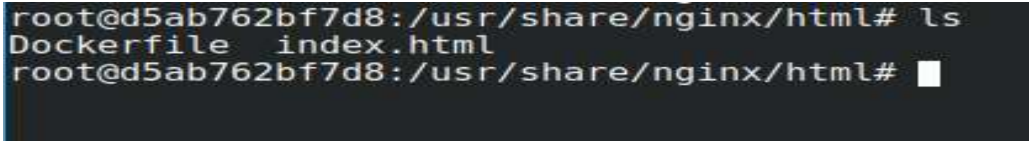


```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker container run -d --name mynginx -p 8080:80 -v $(pwd):/usr/share
/nginx/html nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
Digest: sha256:380eb808e2a3b0dd954f92c1cae2f845e6558a15037efefcabc5b4e03d666d03
Status: Downloaded newer image for nginx:latest
d5ab762bf7d8ae0ac7c985c4c58d53d197ac0a10425a23e3c95965ad0541e1e1
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```

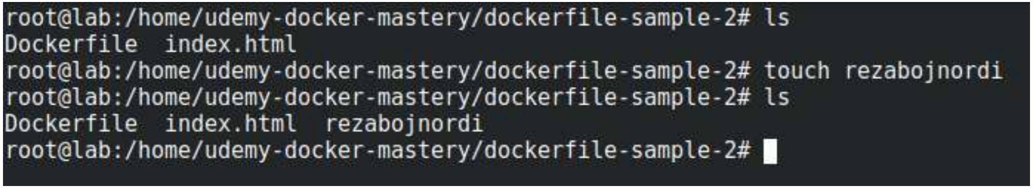


```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# docker container exec -it mynginx bash
root@5b98c157c0f1:/# 
```

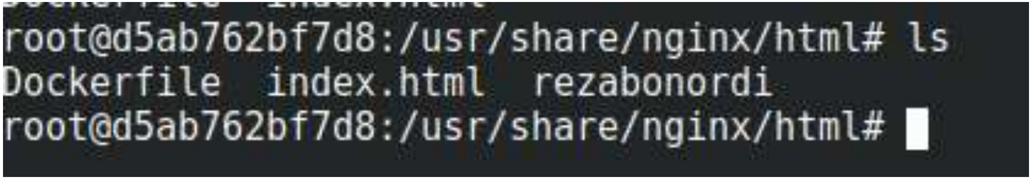
Bind mount



```
root@d5ab762bf7d8:/usr/share/nginx/html# ls
Dockerfile  index.html
root@d5ab762bf7d8:/usr/share/nginx/html# 
```

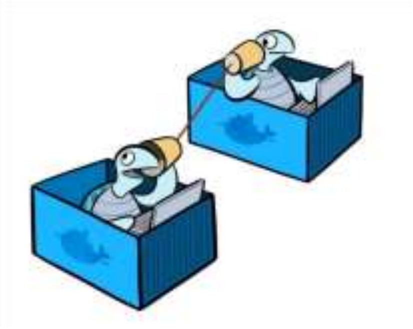


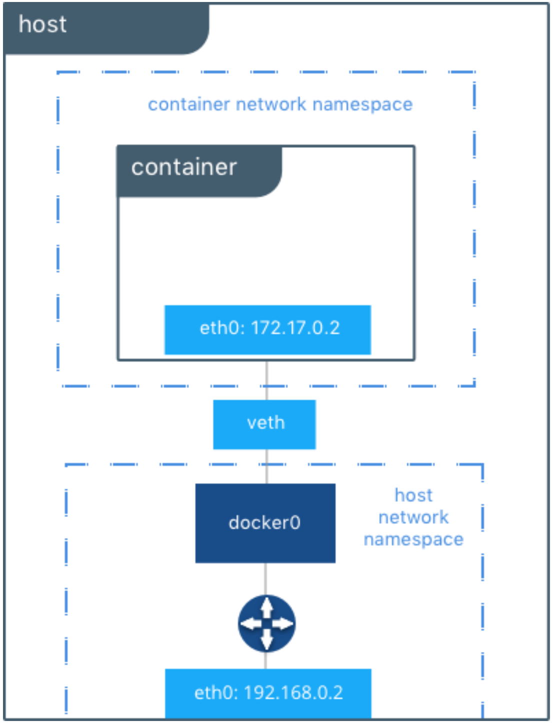
```
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# ls
Dockerfile  index.html
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# touch rezabojnordi
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# ls
Dockerfile  index.html  rezabojnordi
root@lab:/home/udemy-docker-mastery/dockerfile-sample-2# 
```



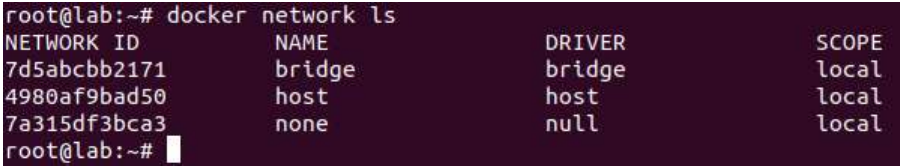
```
root@d5ab762bf7d8:/usr/share/nginx/html# ls
Dockerfile  index.html  rezabonordi
root@d5ab762bf7d8:/usr/share/nginx/html# 
```

Network in docker

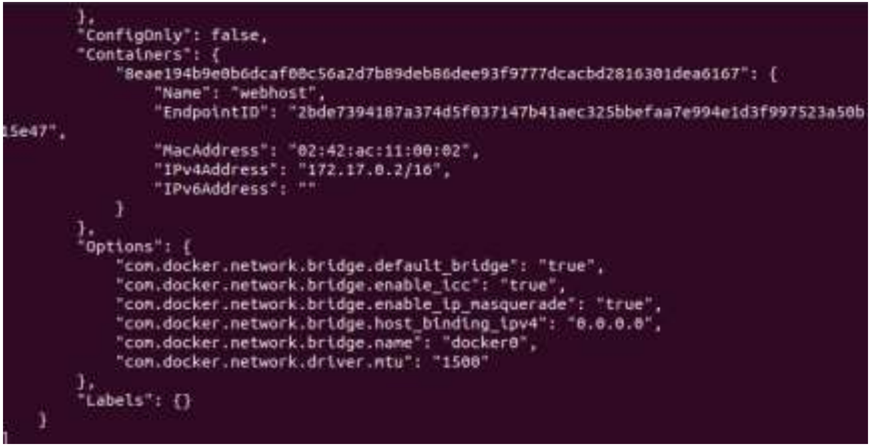




Network in docker

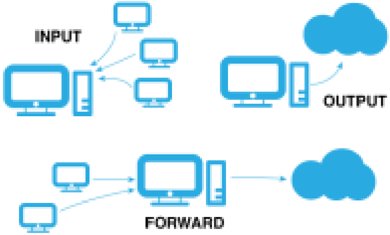


```
root@lab:~# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
7d5abcbb2171        bridge              bridge              local
4980af9bad50        host                host                local
7a315df3bca3        none                null                local
root@lab:~# 
```



```
        },
        "ConfigOnly": false,
        "Containers": {
            "8eae194b9e0b6dcaf00c56a2d7b89deb86dee93f9777dcacbd2816301dea6167": {
                "Name": "webhost",
                "EndpointID": "2bde7394187a374d5f037147b41aec325bbefaa7e994e1d3f997523a50b
15e47",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

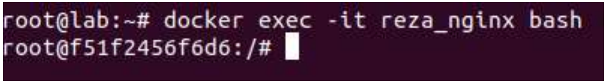
Network in docker



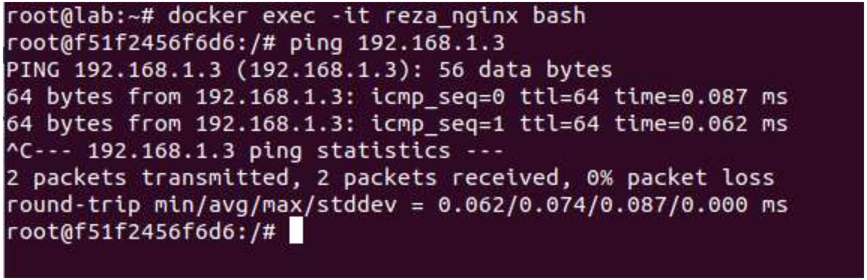
docker network create --subnet=192.168.1.0/24 softgrand_net

docker container run -d --name me_nginx --network softgrand_net nginx

docker container run -d --name reza_nginx --network softgrand_net nginx

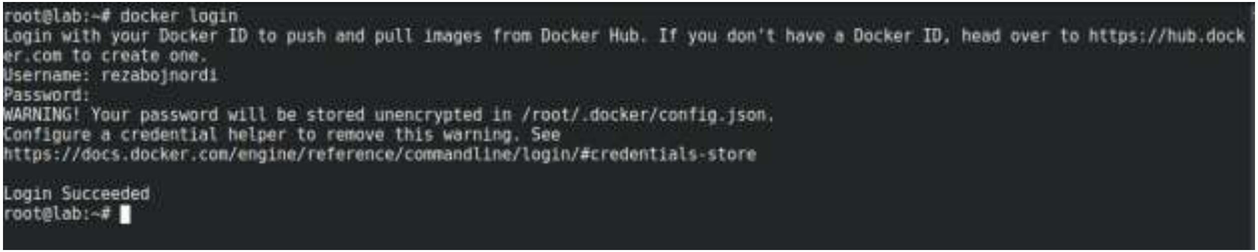


```
root@lab:~# docker exec -it reza_nginx bash
root@f51f2456f6d6:/# 
```

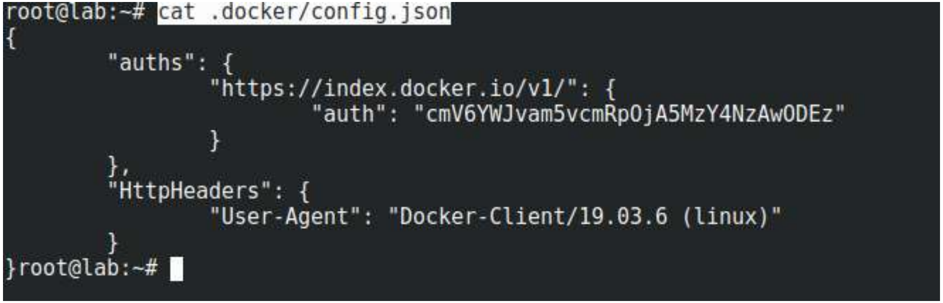


```
root@lab:~# docker exec -it reza_nginx bash
root@f51f2456f6d6:/# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.087 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.062 ms
^C--- 192.168.1.3 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.062/0.074/0.087/0.000 ms
root@f51f2456f6d6:/# 
```

Docker Register

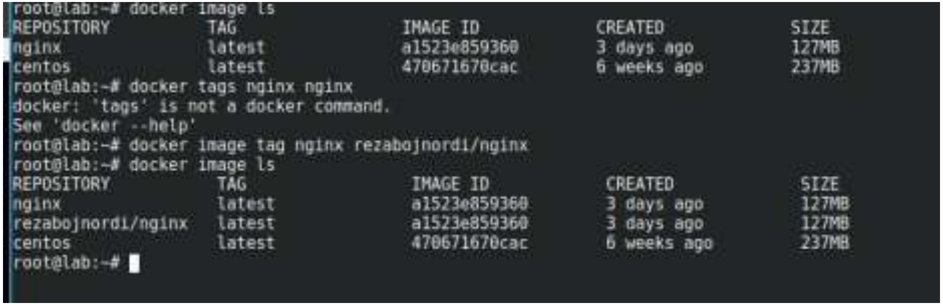


```
root@lab:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.dock
er.com to create one.
Username: rezabojnordi
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@lab:~# 
```

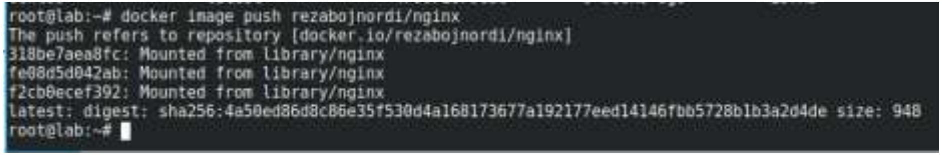


```
root@lab:~# cat .docker/config.json
{
        "auths": {
                "https://index.docker.io/v1/": {
                        "auth": "cmV6YWJvam5vcmRpOjA5MzY4NzAwODEz"
                }
        },
        "HttpHeaders": {
                "User-Agent": "Docker-Client/19.03.6 (linux)"
        }
}root@lab:~# 
```

Docker Register

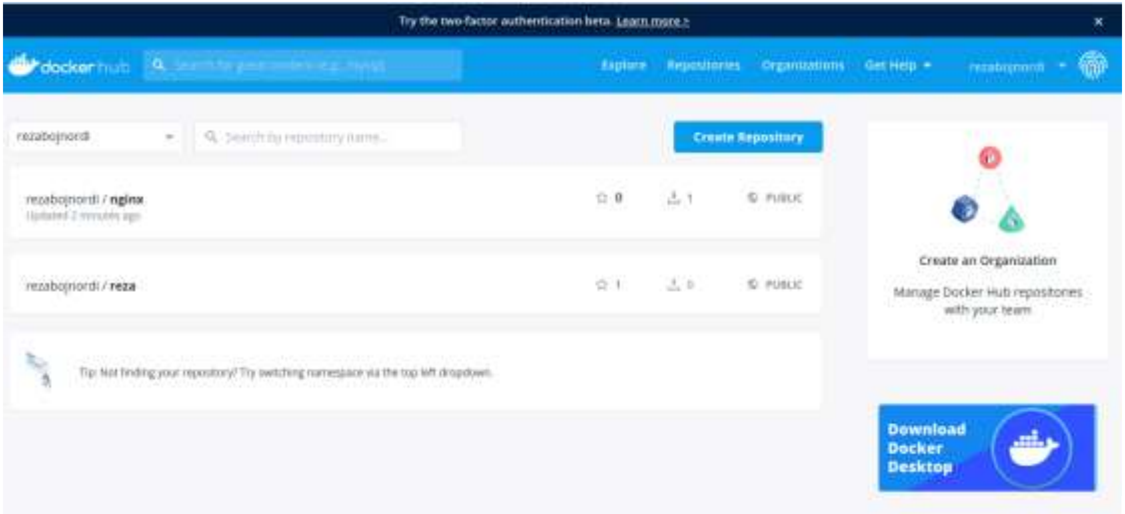


```
root@lab:~# docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              a1523e859360        3 days ago          127MB
centos              latest              470671670cac        6 weeks ago         237MB
root@lab:~# docker tags nginx nginx
docker: 'tags' is not a docker command.
See 'docker --help'
root@lab:~# docker image tag nginx rezabojnordi/nginx
root@lab:~# docker image ls
REPOSITORY           TAG                 IMAGE ID            CREATED             SIZE
nginx                latest              a1523e859360        3 days ago          127MB
rezabojnordi/nginx   latest              a1523e859360        3 days ago          127MB
centos               latest              470671670cac        6 weeks ago         237MB
root@lab:~# █
```



```
root@lab:~# docker image push rezabojnordi/nginx
The push refers to repository [docker.io/rezabojnordi/nginx]
318be7aea8fc: Mounted from library/nginx
fe08d5d042ab: Mounted from library/nginx
f2cb0ecef392: Mounted from library/nginx
latest: digest: sha256:4a50ed86d8c86e35f530d4a168173677a192177eed14146fbb5728b1b3a2d4de size: 948
root@lab:~# █
```

Docker Register



Try the two-factor authentication beta. Learn more >

rezabojnordi / **nginx**
Updated 2 minutes ago
☆ 0 ⤓ 1 PUBLIC

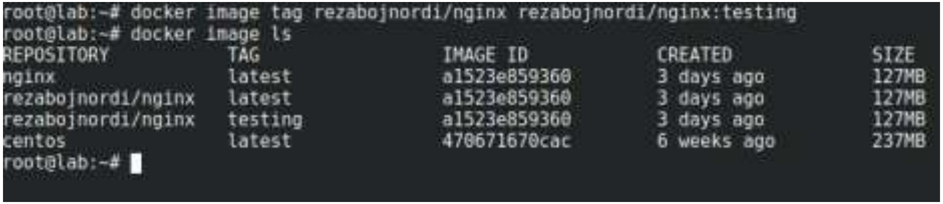
rezabojnordi / **reza**
☆ 1 ⤓ 0 PUBLIC

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

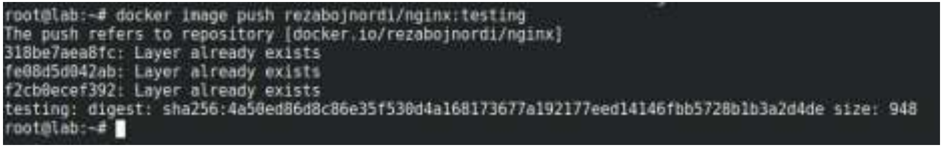
Create an Organization

Manage Docker Hub repositories with your team

Docker Register



```
root@lab:~# docker image tag rezabojnordi/nginx rezabojnordi/nginx:testing
root@lab:~# docker image ls
REPOSITORY           TAG                 IMAGE ID            CREATED             SIZE
nginx                latest              a1523e859360        3 days ago          127MB
rezabojnordi/nginx   latest              a1523e859360        3 days ago          127MB
rezabojnordi/nginx   testing             a1523e859360        3 days ago          127MB
centos               latest              470671670cac        6 weeks ago         237MB
root@lab:~# 
```



```
root@lab:~# docker image push rezabojnordi/nginx:testing
The push refers to repository [docker.io/rezabojnordi/nginx]
318be7aea8fc: Layer already exists
fe08d5d042ab: Layer already exists
f2cb0ecef392: Layer already exists
testing: digest: sha256:4a50ed86d8c86e35f530d4a168173677a192177eed14146fbb5728b1b3a2d4de size: 948
root@lab:~# 
```

Docker Register

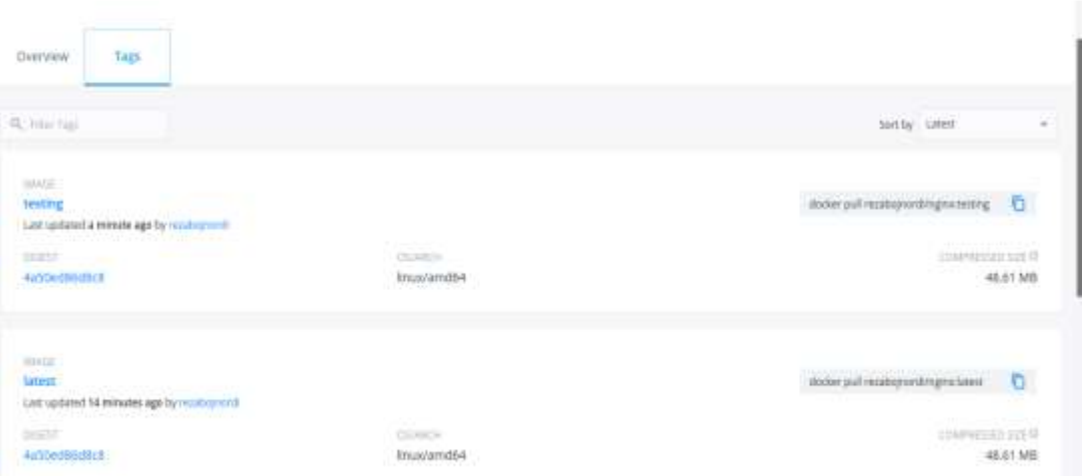


IMAGE
testing
Last updated 4 minute ago by rezabojnordi

```
docker pull rezabojnordi/nginx:testing
```

DIGEST: 4a50ed86d8c8
OS/ARCH: linux/amd64
COMPRESSED SIZE: 48.61 MB

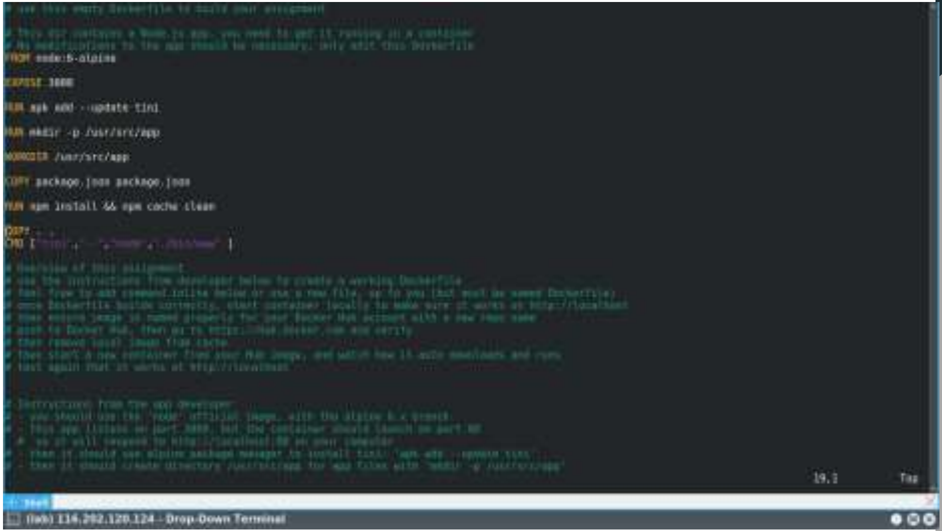
IMAGE
latest
Last updated 14 minutes ago by rezabojnordi

```
docker pull rezabojnordi/nginx:latest
```

DIGEST: 4a50ed86d8c8
OS/ARCH: linux/amd64
COMPRESSED SIZE: 48.61 MB

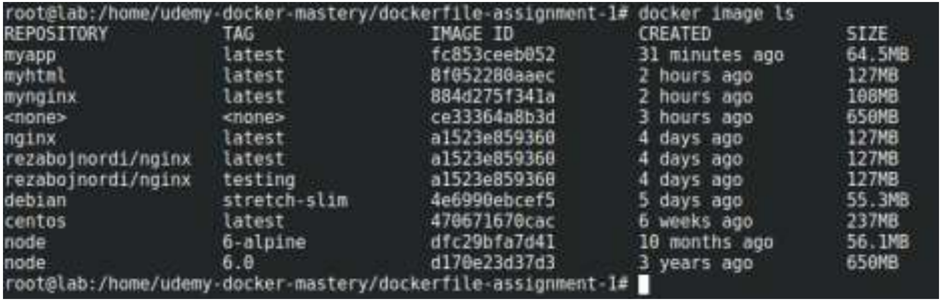
Deploy my app

docker build -t myapp .

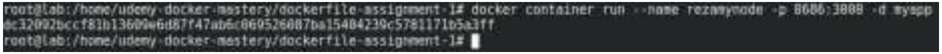


```
# use this empty Dockerfile to build your assignment

# This dir contains a Node.js app, you need to get it running in a container
# No modifications to the app should be necessary, only edit this Dockerfile
FROM node:6-alpine

EXPOSE 3000

RUN apk add --update tini

RUN mkdir -p /usr/src/app

WORKDIR /usr/src/app

COPY package.json package.json

RUN npm install && npm cache clean

COPY . .
CMD ["tini", "--", "node", "./bin/www" ]

# Overview of this assignment
# use the instructions from developer below to create a working Dockerfile
# feel free to add command inline below or use a new file, up to you (but must be named Dockerfile)
# once Dockerfile builds correctly, start container locally to make sure it works on http://localhost
# then ensure image is named properly for your Docker Hub account with a new repo name
# push to Docker Hub, then go to https://hub.docker.com and verify
# then remove local image from cache
# then start a new container from your Hub image, and watch how it auto downloads and runs
# test again that it works at http://localhost


# Instructions from the app developer
# - you should use the 'node' official image, with the alpine 6.x branch
# - this app listens on port 3000, but the container should launch on port 80
#  so it will respond to http://localhost:80 on your computer
# - then it should use alpine package manager to install tini: 'apk add --update tini'
# - then it should create directory /usr/src/app for app files with 'mkdir -p /usr/src/app'
                                                                                19,1          Top
```

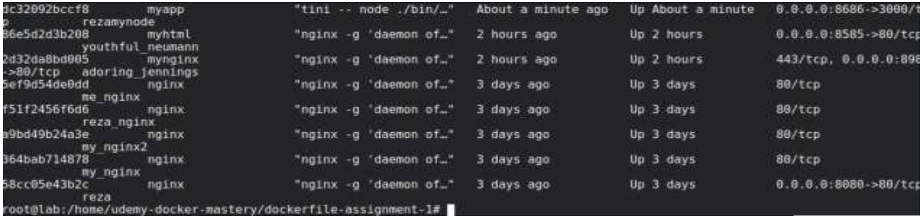
Deploy my app



```
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# docker image ls
REPOSITORY           TAG                 IMAGE ID            CREATED             SIZE
myapp                latest              fc853ceeb052        31 minutes ago      64.5MB
myhtml               latest              8f052280aaec        2 hours ago         127MB
mynginx              latest              884d275f341a        2 hours ago         108MB
<none>               <none>              ce33364a8b3d        3 hours ago         650MB
nginx                latest              a1523e859360        4 days ago          127MB
rezabojnordi/nginx   latest              a1523e859360        4 days ago          127MB
rezabojnordi/nginx   testing             a1523e859360        4 days ago          127MB
debian               stretch-slim        4e6990ebcef5        5 days ago          55.3MB
centos               latest              470671670cac        6 weeks ago         237MB
node                 6-alpine            dfc29bfa7d41        10 months ago       56.1MB
node                 6.0                 d170e23d37d3        3 years ago         650MB
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# 
```

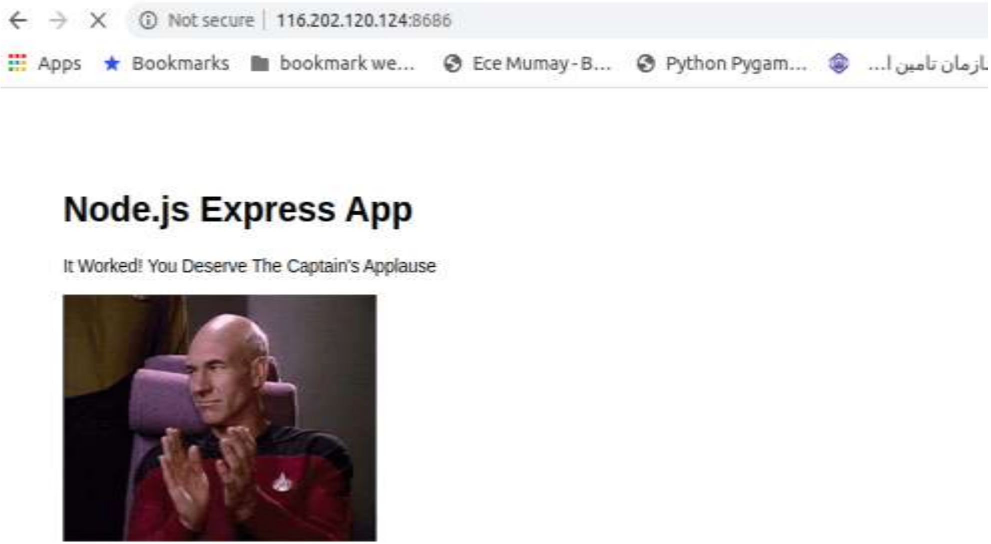


```
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# docker container run --name rezamynode -p 8686:3000 -d myapp
dc32092bccf81b13609e6d87f47ab6c869526087ba15404239c5781171b5a3ff
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# 
```



```
dc32092bccf8        myapp                "tini -- node ./bin/…"  About a minute ago   Up About a minute    0.0.0.0:8686->3000/t
p          rezamynode
86e5d2d3b208        myhtml               "nginx -g 'daemon of…"   2 hours ago          Up 2 hours           0.0.0.0:8585->80/tcp
           youthful_neumann
2d32da8bd005        mynginx              "nginx -g 'daemon of…"   2 hours ago          Up 2 hours           443/tcp, 0.0.0.0:898
->80/tcp   adoring_jennings
5ef9d54de0dd        nginx                "nginx -g 'daemon of…"   3 days ago           Up 3 days            80/tcp
           me_nginx
f51f2456f6d6        nginx                "nginx -g 'daemon of…"   3 days ago           Up 3 days            80/tcp
           reza_nginx
a9bd49b24a3e        nginx                "nginx -g 'daemon of…"   3 days ago           Up 3 days            80/tcp
           my_nginx2
364bab714878        nginx                "nginx -g 'daemon of…"   3 days ago           Up 3 days            80/tcp
           my_nginx
58cc05e43b2c        nginx                "nginx -g 'daemon of…"   3 days ago           Up 3 days            0.0.0.0:8080->80/tcp
           reza
root@lab:/home/udemy-docker-mastery/dockerfile-assignment-1# 
```

Deploy my app

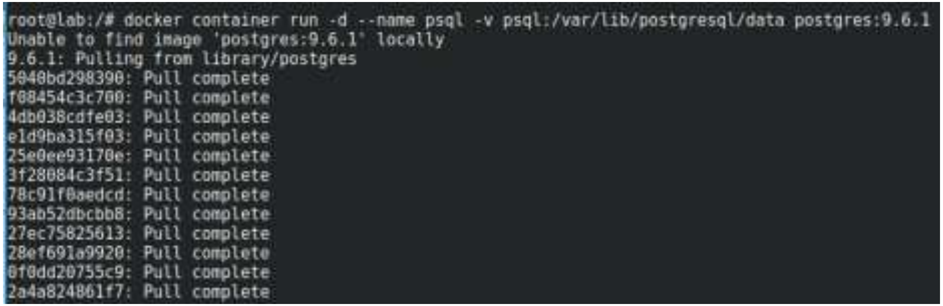


116.202.120.124:8686

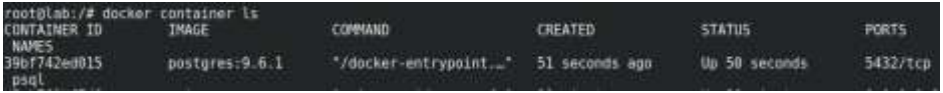
Node.js Express App

It Worked! You Deserve The Captain's Applause

Go from one version to another



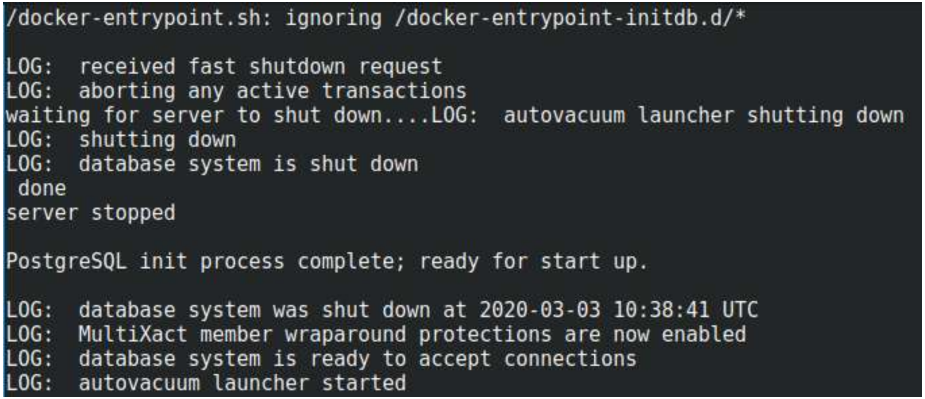
```
root@lab:/# docker container run -d --name psql -v psql:/var/lib/postgresql/data postgres:9.6.1
Unable to find image 'postgres:9.6.1' locally
9.6.1: Pulling from library/postgres
5040bd298390: Pull complete
f08454c3c700: Pull complete
4db038cdfe03: Pull complete
e1d9ba315f03: Pull complete
25e0ee93170e: Pull complete
3f28084c3f51: Pull complete
78c91f0aedcd: Pull complete
93ab52dbcbb8: Pull complete
27ec75825613: Pull complete
28ef691a9920: Pull complete
0f0dd20755c9: Pull complete
2a4a824861f7: Pull complete
```



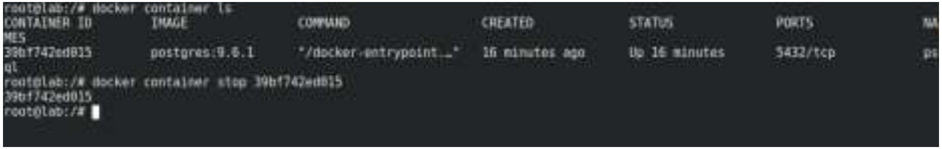
```
root@lab:/# docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
 NAMES
39bf742ed015        postgres:9.6.1      "/docker-entrypoint.…"   51 seconds ago      Up 50 seconds       5432/tcp
 psql
```

Go from one version to another

docker container logs -f psql

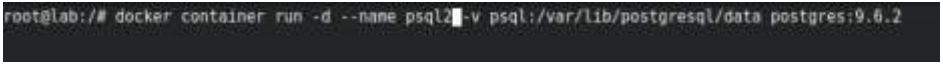


```
/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*

LOG:  received fast shutdown request
LOG:  aborting any active transactions
waiting for server to shut down....LOG:  autovacuum launcher shutting down
LOG:  shutting down
LOG:  database system is shut down
 done
server stopped

PostgreSQL init process complete; ready for start up.

LOG:  database system was shut down at 2020-03-03 10:38:41 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
```

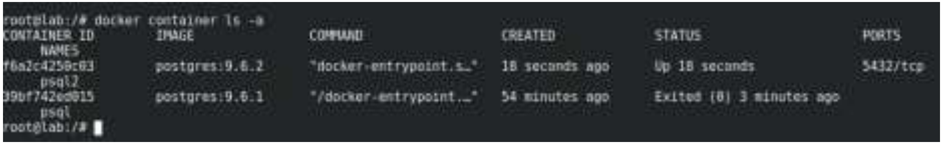


```
root@lab:/# docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NA
MES
39bf742ed015        postgres:9.6.1      "/docker-entrypoint.…"   16 minutes ago      Up 16 minutes       5432/tcp            ps
ql
root@lab:/# docker container stop 39bf742ed015
39bf742ed015
root@lab:/# █
```

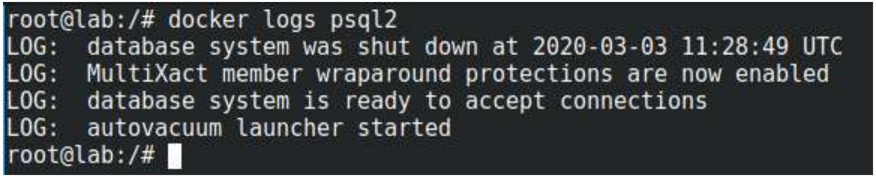
Go from one version to another



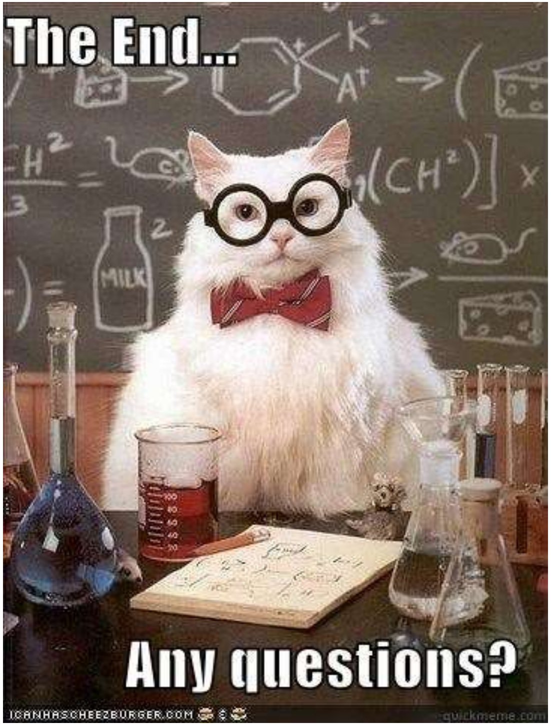
```
root@lab:/# docker container run -d --name psql2 -v psql:/var/lib/postgresql/data postgres:9.6.2
```



```
root@lab:/# docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                   PORTS
     NAMES
f6a2c4250c03        postgres:9.6.2      "docker-entrypoint.s…"   18 seconds ago      Up 18 seconds            5432/tcp
     psql2
39bf742ed015        postgres:9.6.1      "/docker-entrypoint.…"   54 minutes ago      Exited (0) 3 minutes ago
     psql
root@lab:/#
```



```
root@lab:/# docker logs psql2
LOG:  database system was shut down at 2020-03-03 11:28:49 UTC
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
root@lab:/#
```


The End...
Any questions?
ICANHASCHEEZBURGER.COM
quickmeme.com