

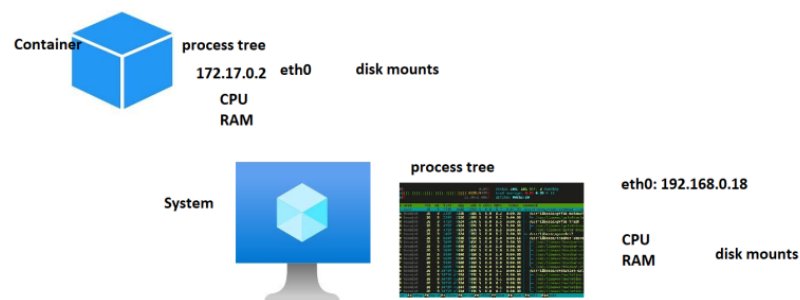


APRIL 8, 2023

DevOps Classroomnotes 08/Apr/2023

How Isolations are created or How Containers Work

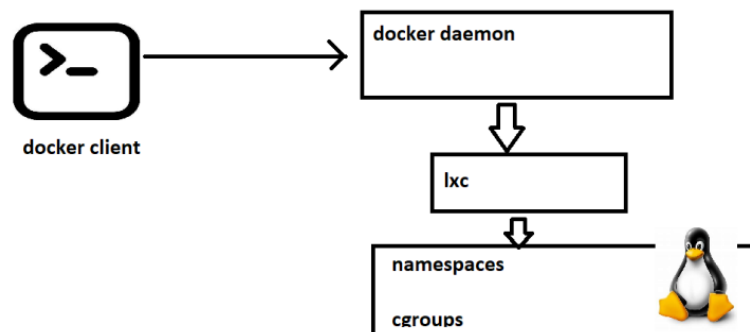
- Each container is getting a
 - new process tree
 - disk mounts
 - network (nic)
 - cpu/memory
 - users
- [Refer Here](#) for Docker Internals



Docker Architecture

Generation 1:

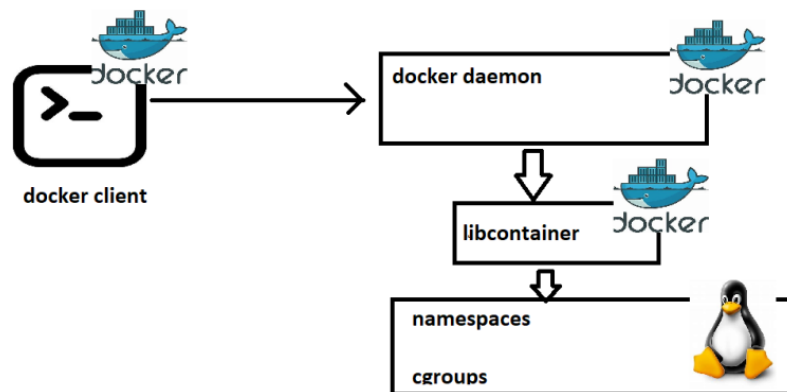
- This was first gen, Where docker daemon used lxc (a linux kernel feature) to create containers



Generation2:

- Since docker was relying on lxc which was kernel feature, updates to kernel frequently used to break containers created by docker.
- So docker has created its own component called libcontainer (libc) to create containers.

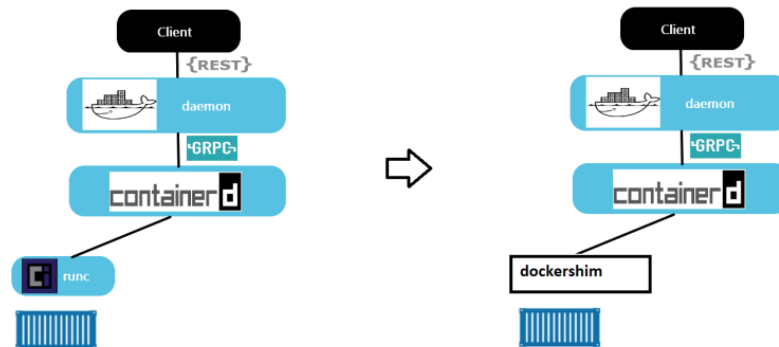
- Docker wanted containers to be multi os and lxc was definitely not the way forward.



- Adoption of docker was drastically increased as it was stable.

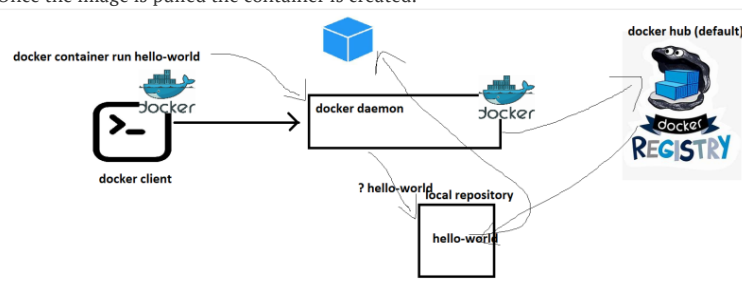
Generation 3:

- In this generation, docker engine was revamped from monolith to multi component architecture and the images and containers were according to OCI (open container initiative) image spec and runtime spec.
- In the latest architecture
- docker daemon exposes api's to listen requests from docker client.
- Passes the requests to containerd. This manages the lifecycle of container
- containerd forks a runc process which creates container. once the container is created the parent of the container will be docker shim



Creating our first docker container

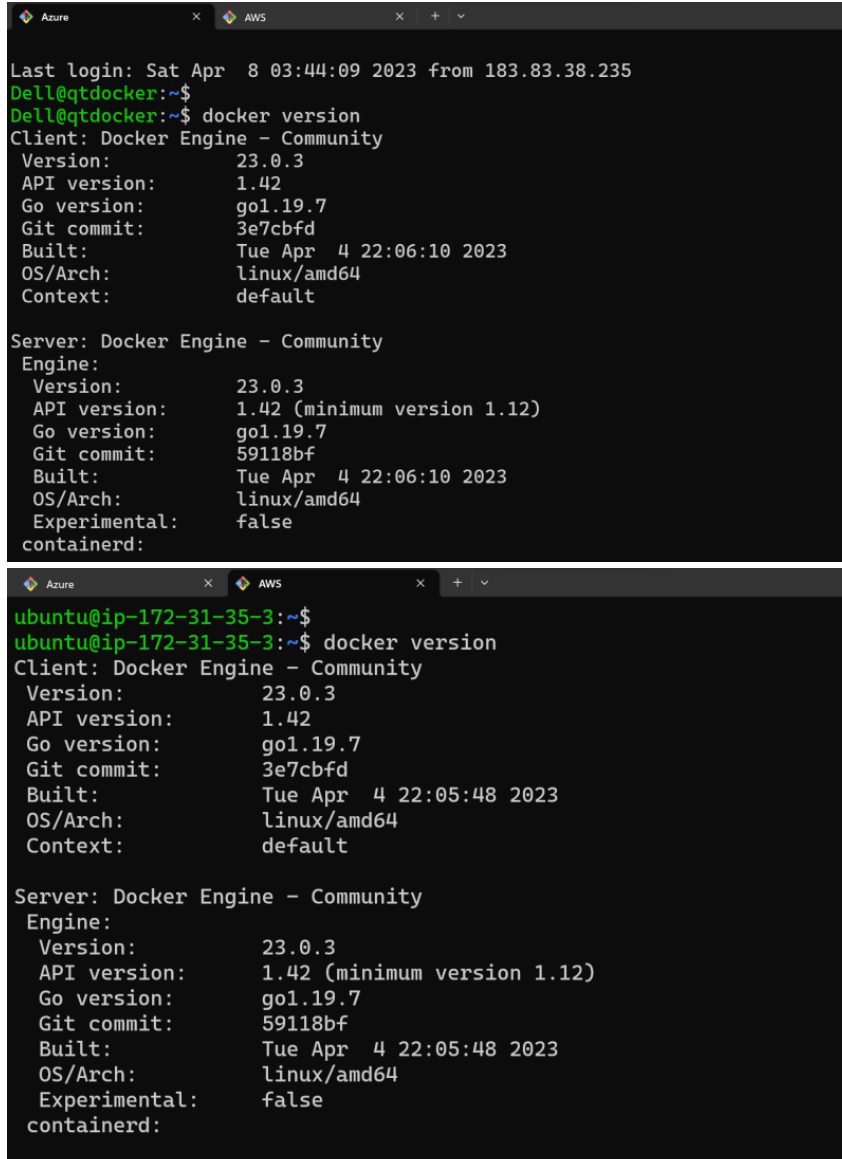
- docker container creation:
- To create container we need some image in this case lets take hello-world
- The command `docker container run hello-world` executed
- What happens
 - docker client will forward the request to docker daemon
 - docker daemon will check if the image exists locally. if yes creates the container by using image
 - if the image doesnot exist, then docker daemon tries to download the image from docker registry connected. The default docker registry is docker hub.
 - Downloading image into local repo from registry is called as pull.
 - Once the image is pulled the container is created.



- Registry is collection of docker images hosted for reuse.
- Docker hub [Refer Here](#)

Playing with containers

- Create a new linux vm and install docker in it



The image shows two terminal windows. The top window is an Azure VM named 'Dell@qtdocker' with IP 183.83.38.235. It shows the output of 'docker version' and 'docker engine' commands, indicating Docker Engine - Community is installed with version 23.0.3. The bottom window is an AWS VM named 'ubuntu@ip-172-31-35-3' with IP 172.31.35.3. It also shows the output of 'docker version' and 'docker engine' commands, indicating Docker Engine - Community is installed with version 23.0.3.

```

Last login: Sat Apr  8 03:44:09 2023 from 183.83.38.235
Dell@qtdocker:~$
Dell@qtdocker:~$ docker version
Client: Docker Engine - Community
Version: 23.0.3
API version: 1.42
Go version: go1.19.7
Git commit: 3e7cbfd
Built: Tue Apr  4 22:06:10 2023
OS/Arch: linux/amd64
Context: default

Server: Docker Engine - Community
Engine:
Version: 23.0.3
API version: 1.42 (minimum version 1.12)
Go version: go1.19.7
Git commit: 59118bf
Built: Tue Apr  4 22:06:10 2023
OS/Arch: linux/amd64
Experimental: false
containerd:

```

```

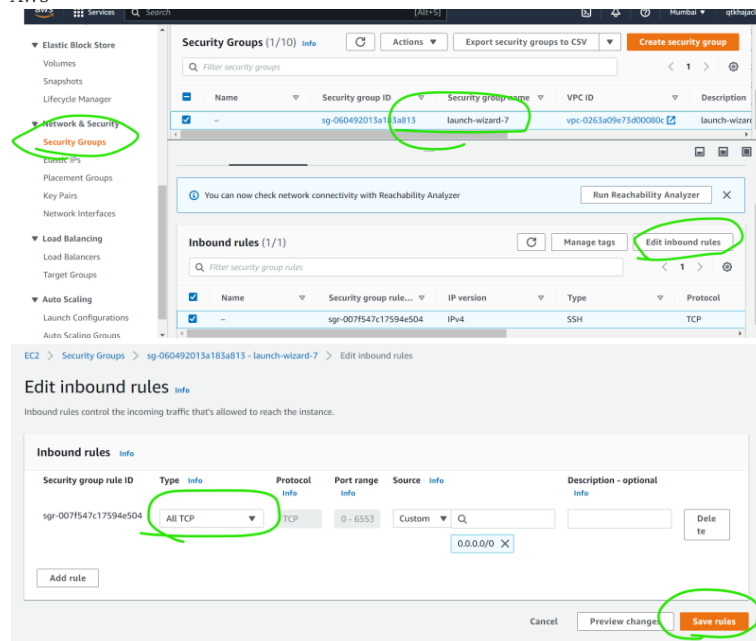
ubuntu@ip-172-31-35-3:~$
ubuntu@ip-172-31-35-3:~$ docker version
Client: Docker Engine - Community
Version: 23.0.3
API version: 1.42
Go version: go1.19.7
Git commit: 3e7cbfd
Built: Tue Apr  4 22:05:48 2023
OS/Arch: linux/amd64
Context: default

Server: Docker Engine - Community
Engine:
Version: 23.0.3
API version: 1.42 (minimum version 1.12)
Go version: go1.19.7
Git commit: 59118bf
Built: Tue Apr  4 22:05:48 2023
OS/Arch: linux/amd64
Experimental: false
containerd:

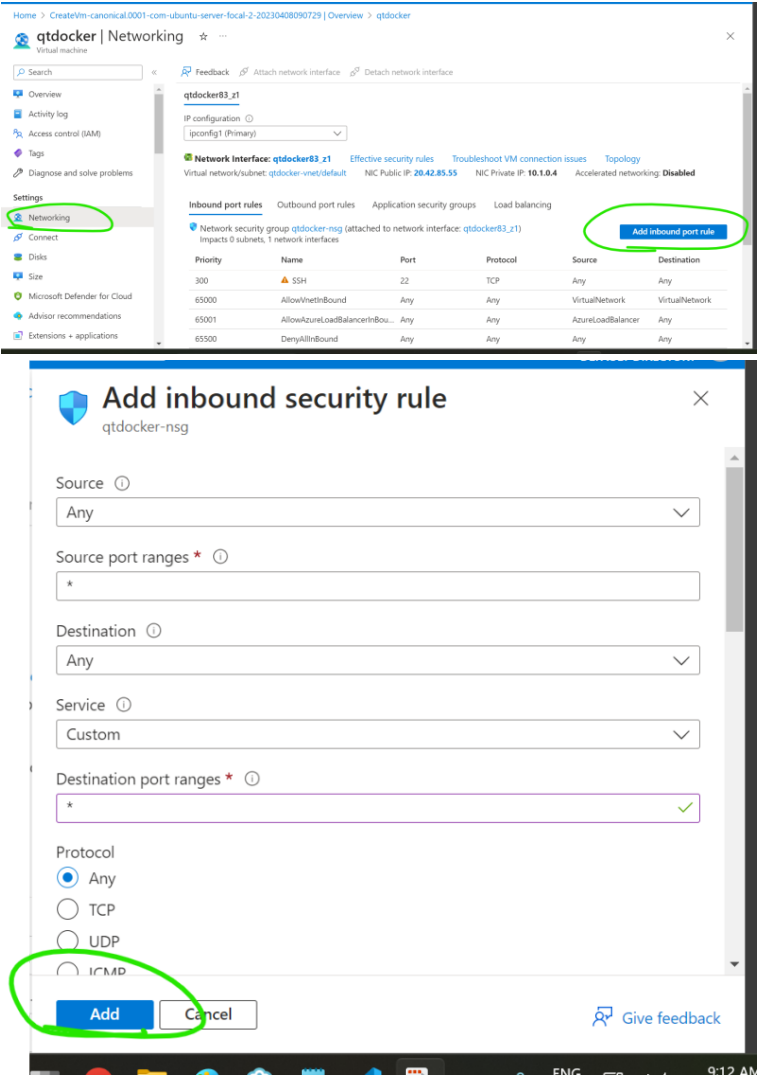
```

- Open all the ports

- AWS



• Azure



Check docker images in the host

```

Dell@qtdocker:~$ docker image --help

Usage:  docker image COMMAND

Manage images

Commands:
  build      Build an image from a Dockerfile
  history    Show the history of an image
  import     Import the contents from a tarball to create a filesystem image
  inspect    Display detailed information on one or more images
  load       Load an image from a tar archive or STDIN
  ls         List images
  prune      Remove unused images
  pull       Download an image from a registry
  push       Upload an image to a registry
  rm         Remove one or more images
  save       Save one or more images to a tar archive (streamed to STDOUT by default)
  tag        Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
Dell@qtdocker:~$

Dell@qtdocker:~$ docker image ls --help

Usage:  docker image ls [OPTIONS] [REPOSITORY[:TAG]]

List images

Aliases:
  docker image ls, docker image list, docker images

Options:
  -a, --all          Show all images (default hides intermediate images)
  --digests          Show digests
  -f, --filter filter Filter output based on conditions provided
  --format string     Format output using a custom template:
                     'table':          Print output in table format with column
                     headers (default)
                     'table TEMPLATE': Print output in table format using the
                     given Go template
                     'json':          Print in JSON format
                     'TEMPLATE':      Print output using the given Go template.
                     Refer to https://docs.docker.com/go/formatting/ for more
                     information about formatting output with templates
  --no-trunc         Don't truncate output
  -q, --quiet         Only show image IDs
Dell@qtdocker:~$

ubuntu@ip-172-31-35-3:~$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu@ip-172-31-35-3:~$

```

pull the images from docker hub

- image naming convention

```

[username]/[repository]:[<tag>]
shaikkhajaibrahim/myspc:1.0.1
username => shaikkhajaibrahi
repository => what image => myspc
tag => version => 1.0.1

```

- default tag is latest

```

nginx
nginx:latest

```

- official images dont have username

```

nginx
ubuntu
alpine
shaikkhajaibrahim/myspc

```

- Lets pull the image nginx with tag 1.23

```

docker image pull nginx:1.23
docker image ls

```

```
Azure x AWS x + v
ubuntu@ip-172-31-35-3:~$ docker image pull nginx:1.23
1.23: Pulling from library/nginx
f1f26f570256: Pull complete
7f7f30930c6b: Pull complete
2836b727df80: Pull complete
e1eeb0f1c06b: Pull complete
86b2457cc2b0: Pull complete
9862f2ee2e8c: Pull complete
Digest: sha256:2ab30d6ac53580a6db8b657abf0f68d75360ff5cc1670a85acb5bd85ba1b19c0
Status: Downloaded newer image for nginx:1.23
docker.io/library/nginx:1.23
ubuntu@ip-172-31-35-3:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         1.23      080ed0ed8312   10 days ago    142MB
ubuntu@ip-172-31-35-3:~$
```

* Lets pull the jenkins image with latest version

```
Azure x AWS x + v
Dell@qtdocker:~$ docker image pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
3e440a704568: Pull complete
3ae39743cf11: Pull complete
b44e6bfff74b7: Pull complete
b177c8e31ff5: Pull complete
f01b258d8dd0: Pull complete
9cbc2a872ded: Pull complete
e57a2db25a0c: Pull complete
0a7ec3b29016: Pull complete
3dd80d927a98: Pull complete
8fcaca854d4d: Pull complete
f3eba20d7da1: Pull complete
ba09b87316f0: Pull complete
5b2457263792: Pull complete
Digest: sha256:562e24559714de46f3fe80a001eccc6507e03c1f32f73bbf51f6113eb45ebc8d
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
Dell@qtdocker:~$
```

* Lets find the alpine and pull the image

```
Dell@qtdocker:~$ docker image pull alpine:3.16
3.16: Pulling from library/alpine
91d30c5bc195: Pull complete
Digest: sha256:c2b622f6e510a0d25bccaffa9e67b75a6860cb09b74bb58cfc36a9ef4331109f
Status: Downloaded newer image for alpine:3.16
docker.io/library/alpine:3.16
Dell@qtdocker:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
jenkins/jenkins  latest    e701a1b6fb83   3 days ago     471MB
alpine         3.16      8471affe5de5   9 days ago     5.54MB
Dell@qtdocker:~$
```

Remove images from local

- Every image will have unique image id and image name
- We can delete individually `docker image rm alpine:3.17`
- if i have to delete all the images `docker image rm $(docker image ls -q)`

```
Dell@qtdocker:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
jenkins/jenkins  latest    e701a1b6fb83   3 days ago     471MB
alpine         3.16      8471affe5de5   9 days ago     5.54MB
alpine         3.17      9ed4aefc74f6   9 days ago     7.04MB
Dell@qtdocker:~$ docker image ls -q
e701a1b6fb83
8471affe5de5
9ed4aefc74f6
Dell@qtdocker:~$ docker image rm $(docker image ls -q)
Untagged: jenkins/jenkins:latest
Untagged: jenkins/jenkins@sha256:562e24559714de46f3fe80a001eccc6507e03c1f32f73bbf51f6113eb45ebc8d
Deleted: sha256:e701a1b6fb8349e97b3f25d0ba4e00fb3f9668ccb0acf1
Deleted: sha256:c2b622f6e510a0d25bccaffa9e67b75a6860cb09b74bb58cfc36a9ef4331109f
Dell@qtdocker:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
Dell@qtdocker:~$
```

Create a container with nginx

- To create and start the container we use run command

```
Dell@qtdocker:~$ docker container run -d nginx:1.22
Unable to find image 'nginx:1.22' locally
1.22: Pulling from library/nginx
f1f26f570256: Already exists
fd03b214f774: Pull complete
ef2fc869b944: Pull complete
ac713a9ef2cc: Pull complete
fd071922d543: Pull complete
2a9f38700bb5: Pull complete
Digest: sha256:fc5f5fb7574755c306aaf88456ebf0b006420a184d52b923d2f0197108f6b7
Status: Downloaded newer image for nginx:1.22
340e9bca8bf6610c1061250071bcb825737beba63bad2a6c74d9a0a88924c7dc
Dell@qtdocker:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	P
ORTS	NAMES				
340e9bca8bf6	nginx:1.22	"/docker-entrypoint..."	6 seconds ago	Up 5 seconds	8
0/tcp	mystifying_napier				

```
Dell@qtdocker:~$
```

- note: i will be using -d for some time and we will discuss importance of this in next session
- every container gets an id and a name. name can be passed while creating container, if not docker will give random name

```
Dell@qtdocker:~$ docker container run --name mynginx1 -d nginx:1.22
6531f97fd9c2e4e77607e13c57684c4f14c9c4f5106736a1908c463ec10bc228
Dell@qtdocker:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
6531f97fd9c2	nginx:1.22	"/docker-entrypoint..."	8 seconds ago	Up 8 seconds	80/tcp
mynginx1					
340e9bca8bf6	nginx:1.22	"/docker-entrypoint..."	2 minutes ago	Up 2 minutes	80/tcp
mystifying_napier					

```
Dell@qtdocker:~$
```

- Remove all the running containers `docker container rm -f $(docker container ls -q)`

```
Dell@qtdocker:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
7d7f907d8dfa	nginx:1.22	"/docker-entrypoint..."	5 seconds ago	Up 4 seconds	80/tcp
mynginx3					
27e4c7416ecc	nginx:1.22	"/docker-entrypoint..."	9 seconds ago	Up 7 seconds	80/tcp
mynginx2					
ef62979992ee	nginx:1.22	"/docker-entrypoint..."	14 seconds ago	Up 13 seconds	80/tcp
mynginx1					
340e9bca8bf6	nginx:1.22	"/docker-entrypoint..."	5 minutes ago	Up 5 minutes	80/tcp
mystifying_napier					

```
Dell@qtdocker:~$ docker container ls -q
7d7f907d8dfa
27e4c7416ecc
ef62979992ee
340e9bca8bf6
Dell@qtdocker:~$ docker container rm -f $(docker container ls -q)
7d7f907d8dfa
27e4c7416ecc
ef62979992ee
340e9bca8bf6
Dell@qtdocker:~$
```

```
Dell@qtdocker:~$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
Dell@qtdocker:~$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
Dell@qtdocker:~$
```

- Remove specific container

```
Dell@qtdocker:~$ docker container rm mynginx1
Error response from daemon: You cannot remove a running container 6531f97fd9c2e4e77607e13c57684c4f14c9c4f5106736a1908c463ec10bc228. Stop the container before attempting removal or force remove
Dell@qtdocker:~$ docker container rm -f mynginx1
mynginx1
Dell@qtdocker:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
340e9bca8bf6	nginx:1.22	"/docker-entrypoint..."	4 minutes ago	Up 4 minutes	80/tcp
mystifying_napier					

```
Dell@qtdocker:~$
```

- Remove all containers `docker container rm -f $(docker container ls -a -q)`

```
Dell@qtdocker:~$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d983e8b31c2b nginx:1.22 "/docker-entrypoint..." 7 seconds ago Up 6 seconds 80/tcp mynginx3
8401c1b26a08 nginx:1.22 "/docker-entrypoint..." 14 seconds ago Up 13 seconds 80/tcp mynginx2
778c25cd3830 nginx:1.22 "/docker-entrypoint..." 20 seconds ago Up 19 seconds 80/tcp mynginx1
Dell@qtdocker:~$ docker container stop mynginx2
mynginx2
Dell@qtdocker:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d983e8b31c2b	nginx:1.22	"/docker-entrypoint..."	28 seconds ago	Up 28 seconds	80/tcp	mynginx3
778c25cd3830	nginx:1.22	"/docker-entrypoint..."	41 seconds ago	Up 41 seconds	80/tcp	mynginx1

```
Dell@qtdocker:~$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
d983e8b31c2b nginx:1.22 "/docker-entrypoint..." 43 seconds ago Up 42 seconds
80/tcp mynginx3
Dell@qtdocker:~$
778c25cd3830 nginx:1.22 "/docker-entrypoint..." 56 seconds ago Up 55 seconds
80/tcp mynginx1
Dell@qtdocker:~$
```

- Exercise: Start and stop containers

Leave a Reply

Enter your comment here...

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)



About continuous learner

devops & cloud enthusiastic learner

[VIEW ALL POSTS](#)

◀ [PREVIOUS POST](#)

[AWS Classroomnotes 07/Apr/2023](#)

[NEXT POST](#)

Terraform Book

POWERED BY [WORDPRESS.COM](#).