

# Table of contents

- [Table of contents](#)
    - [Docker Installation](#)
      - [Install Docker on EC2 \(Manually\)](#)
      - [Install Docker on EC2 \(via EC2 User data\)](#)
    - [Docker Architecture](#)
    - [Docker Commands](#)
    - [Docker Container Lifecycle Commands](#)
      - [Create Containers](#)
      - [Start Container](#)
      - [Run Container](#)
      - [Stop Container](#)
      - [Restart Container](#)
      - [Delete Container](#)
    - [Changes inside Container Files](#)
- 

## Docker Installation

### Install Docker on EC2 (Manually)

- Create an EC2 instance with Amazon Linux 2 & execute these commands after you SSH into it.

```
sudo yum update -y
sudo amazon-linux-extras install docker -y
docker -v
```

- Above commands will install Docker and Start the docker service/daemon.
- Start the docker service and verify the status of Docker Daemon

```
sudo systemctl status docker
sudo systemctl start docker
sudo systemctl enable docker
```

--

```

[ec2-user@ip-172-31-1-197 ~]$ sudo systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-07-17 06:39:46 UTC; 3h 36min ago
     Docs: https://docs.docker.com
    Process: 7628 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 7618 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 7634 (dockerd)
      Tasks: 9
     Memory: 205.7M
    CGroup: /system.slice/docker.service
            └─7634 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nfile=1024:4096

Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.075372236Z" level=info msg="ClientConn sw...=grpc
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.125677984Z" level=info msg="Loading conta...art."
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.311067572Z" level=info msg="Default bridg...ress"
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.369856229Z" level=info msg="Loading conta...one."
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.386963306Z" level=info msg="Docker daemon...10.4
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.387468167Z" level=info msg="Daemon has co...tion"
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal systemd[1]: Started Docker Application Container Engine.
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.415505904Z" level=info msg="API listen on...sock"
Jul 17 07:02:27 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T07:02:27.940196427Z" level=info msg="ignoring even...lete"
Jul 17 10:16:17 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T10:16:17.625481480Z" level=info msg="ignoring even...lete"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-1-197 ~]$

```

--

- After installation of docker service, a Linux Group **docker** is created.
- Add **ec2-user** to the docker group to execute the docker commands in linux with **ec2-user**

```

docker info
cat /etc/group | grep docker
sudo usermod -a -G docker ec2-user
cat /etc/group | grep docker
docker info

```

- Logout from the EC2 instance and log back in or restart the ssh session and execute **docker images**

## Install Docker on EC2 (via EC2 User data)

- Pass below commands via EC2 User Data while launching the EC2 Instance

```

#!/bin/sh
yum update -y
amazon-linux-extras install docker -y
service docker start
usermod -a -G docker ec2-user
chkconfig docker on

```

### ▼ Advanced Details

User data ⓘ

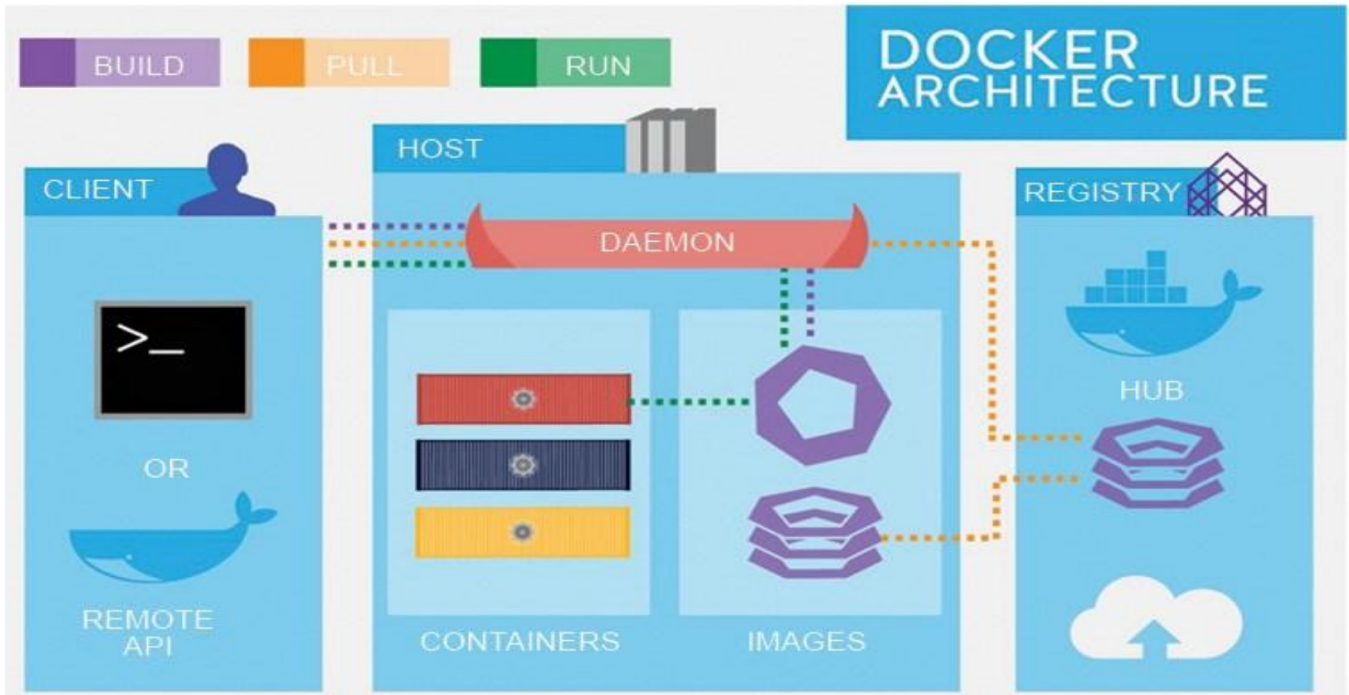
☒ As text ☐ As file ☐ Input is already base64 encoded

```

#!/bin/sh
yum update -y
amazon-linux-extras install docker
service docker start
usermod -a -G docker ec2-user
chkconfig docker on

```

## Docker Architecture



## Docker Commands

```
docker images
# Below command will pull the image from Docker Hub into Local Machine
docker pull ubuntu:18.04
docker images
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
[ec2-user@ip-172-31-20-228 ~]$ docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
2f94e549220a: Pull complete
Digest: sha256:37b7471c1945a2a12e5a57488ee4e3e216a8369d0b9ee1ec2e41db9c2c1e3d22
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
[ec2-user@ip-172-31-20-228 ~]$ docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
ubuntu          18.04           886eca19e611    2 weeks ago     63.1MB
[ec2-user@ip-172-31-20-228 ~]$
```

- [https://hub.docker.com/\\_/ubuntu/tags](https://hub.docker.com/_/ubuntu/tags)

--

```
docker images
# creates and starts a container, allocates a pseudo-TTY connected to the
container's stdin, & creates an interactive bash shell in the container.
docker run -i -t ubuntu:18.04 /bin/bash
hostname
id ec2-user
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker run -i -t ubuntu:18.04 /bin/bash
root@7095aca78d63:/# id
uid=0(root) gid=0(root) groups=0(root)
root@7095aca78d63:/# hostname
7095aca78d63
root@7095aca78d63:/# id ec2-user
id: 'ec2-user': no such user
root@7095aca78d63:/#
```

- Here the SSH Session from EC2 instance is changed to a bash session inside the Docker Container.
- To Detach docker from interactive container, enter the **Ctrl + P** and **Ctrl + Q** escape sequence.

This escape sequence will detach the TTY from the container and land us in the Docker host prompt \$, however the container will continue to run.

```
# Display the running Containers
docker ps
docker run -i -t ubuntu:20.04 /bin/bash
# docker run command first checks if docker image is available locally, if
required image is not present, it pulls the image from Docker Hub and then creates
a new container
id ec2-user
# enter the `Ctrl + P and Ctrl + Q` escape sequence
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker run -i -t ubuntu:18.04 /bin/bash
root@7095aca78d63:/# id
uid=0(root) gid=0(root) groups=0(root)
root@7095aca78d63:/# hostname
7095aca78d63
root@7095aca78d63:/# id ec2-user
id: 'ec2-user': no such user
root@7095aca78d63:/# [ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
7095aca78d63   ubuntu:18.04   "/bin/bash"             3 minutes ago  Up 3 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$ docker run -i -t ubuntu:20.04 /bin/bash
Unable to find image 'ubuntu:20.04' locally
20.04: Pulling from library/ubuntu
ea362f368469: Pull complete
Digest: sha256:b5a61709a9a44284d88fb12e5c48db0409cfad5b69d4ff8224077c57302df9cf
Status: Downloaded newer image for ubuntu:20.04
root@ade7b6905ca2:/# id ec2-user
id: 'ec2-user': no such user
root@ade7b6905ca2:/# [ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
ade7b6905ca2   ubuntu:20.04   "/bin/bash"             About a minute ago  Up About a minute                objective_brattain
7095aca78d63   ubuntu:18.04   "/bin/bash"             9 minutes ago     Up 8 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$
```

```
docker ps
# Create a new container with a specific name
docker run --name=myAmznLinuxContainer -i -t amazonlinux:latest /bin/bash
id
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker run --name=myAmznLinuxContainer -i -t amazonlinux:latest /bin/bash
Unable to find image 'amazonlinux:latest' locally
latest: Pulling from library/amazonlinux
3a461b3ae562: Pull complete
Digest: sha256:ffc013f79b36a2c0352b444f5322ff43de25152a8ac1ad0fa473e90f1cbedcbe
Status: Downloaded newer image for amazonlinux:latest
bash-4.2# id
uid=0(root) gid=0(root) groups=0(root)
bash-4.2# hostname
bash: hostname: command not found
bash-4.2#
```

# enter the `Ctrl + P and Ctrl + Q` escape sequence

docker images

docker ps

```
[ec2-user@ip-172-31-20-228 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
amazonlinux   latest    30156a0e4f04   9 days ago    164MB
ubuntu        20.04     d13c942271d6   2 weeks ago   72.8MB
ubuntu        18.04     886eca19e611   2 weeks ago   63.1MB
[ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND          CREATED        STATUS        PORTS          NAMES
99a0be2ab835   amazonlinux:latest  "/bin/bash"      2 minutes ago  Up 2 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04      "/bin/bash"      9 minutes ago  Up 9 minutes                objective_brattain
7095aca78d63   ubuntu:18.04      "/bin/bash"      16 minutes ago Up 16 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$
```

--

- Connect back to container prompt using `docker attach`

docker ps

# Syntax of below command is :

# docker attach CONTAINER\_NAME|CONTAINER\_ID

docker attach myAmznLinuxContainer

echo \$HOME

```
[ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND          CREATED        STATUS        PORTS          NAMES
99a0be2ab835   amazonlinux:latest  "/bin/bash"      7 minutes ago  Up 7 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04      "/bin/bash"      14 minutes ago Up 14 minutes                objective_brattain
7095aca78d63   ubuntu:18.04      "/bin/bash"      21 minutes ago Up 21 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$ docker attach myAmznLinuxContainer
bash-4.2# echo $HOME
/root
bash-4.2#
```

--

- Search for Docker Image in Docker Hub Registry

docker search amazonlinux

```
[ec2-user@ip-172-31-20-228 ~]$ docker search amazonlinux
NAME                                DESCRIPTION                                STARS    OFFICIAL    AUTOMATED
amazonlinux                        Amazon Linux provides a stable, secure, and ... 1139     [OK]
amd64/amazonlinux                  Amazon Linux provides a stable, secure, and ... 3
hayajo/amazonlinux-sshd             https://github.com/hayajo/amazonlinux-sshd/    3        [OK]
digitalgenius/amazonlinux-python3  2
arm64v8/amazonlinux                Amazon Linux provides a stable, secure, and ... 1
brandedholdings/amazonlinux         Customized amazonlinux build                  1
michaukrieg/amazonlinux-occt-cgal-node Docker image of amazonlinux with opencascade... 1
dokken/amazonlinux-2                Amazon Linux 2 image for kitchen-dokken        1
n0ts/amazonlinux-xray              My AmazonLinux X-Ray Docker Image             1        [OK]
blutv/amazonlinux-python27          0
abeja/amazonlinux-elixir-node-awscli [DEPRECATED] The docker image based on Amazo... 0        [OK]
troллин/amazonlinux                0
ideavate/amazonlinux-node           Docker image starting with Amazon Linux 2 an... 0        [OK]
socrateessystems/amazonlinux        amazonlinux /w aws cli                        0        [OK]
dokken/amazonlinux                  Amazon Linux 201X image for kitchen-dokken      0
taishin/amazonlinux-sshd            0
robertdebock/amazonlinux            Container to test Ansible roles in, includin... 0
blutv/amazonlinux2-python3          0
testbeds/amazonlinux                Amazon Linux                                  0        [OK]
onema/amazonlinux-numpy-lambda      DEPREACATED, USE https://hub.docker.com/r/on... 0
olisimon/amazonlinux-phoenix         Image based on amazonlinux for building or r... 0
diodonfrost/amazonlinux-2-ansible   DEPRECATED - move to diodonfrost/ansible-ama... 0        [OK]
yasuhiroki/amazonlinux-rails        0
enicia/amazonlinux-ruby              Ruby docker container based on AmazonLinux     0
craftmachine/amazonlinux-python3-terraform 0
[ec2-user@ip-172-31-20-228 ~]$
```

--

## Docker Container Lifecycle Commands

### Create Containers

- Using the **docker create** command will create a new Docker container with the specified docker image.

```
docker ps -a
docker create -i --name=myTestContainer -t ubuntu:20.04 /bin/bash
docker ps -a
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS   NAMES
99a0be2ab835   amazonlinux:latest   "/bin/bash"             15 minutes ago Up 15 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04         "/bin/bash"             22 minutes ago Up 22 minutes                objective_brattain
7095aca78d63   ubuntu:18.04         "/bin/bash"             29 minutes ago Up 29 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$ docker create --name=myTestContainer ubuntu:20.04
a3bea9592de41fe2353093064666086e1bc68768cfc6d9bc06e34f3bfeee3804
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS   NAMES
a3bea9592de4   ubuntu:20.04         "bash"                  2 seconds ago Created                myTestContainer
99a0be2ab835   amazonlinux:latest   "/bin/bash"             15 minutes ago Up 15 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04         "/bin/bash"             22 minutes ago Up 22 minutes                objective_brattain
7095aca78d63   ubuntu:18.04         "/bin/bash"             30 minutes ago Up 29 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$
```

--

### Start Container

- To start a stopped container OR to start a container that is having a **Created** Status.

```
docker start myTestContainer
docker ps -a
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ceeb928f742f	ubuntu:20.04	"/bin/bash"	4 seconds ago	Created		myTestContainer
99a0be2ab835	amazonlinux:latest	"/bin/bash"	21 minutes ago	Up 21 minutes		myAmznLinuxContainer
ade7b6905ca2	ubuntu:20.04	"/bin/bash"	28 minutes ago	Up 28 minutes		objective_brattain
7095aca78d63	ubuntu:18.04	"/bin/bash"	35 minutes ago	Up 35 minutes		nifty_mcclintock

```
[ec2-user@ip-172-31-20-228 ~]$ docker start myTestContainer
myTestContainer
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ceeb928f742f	ubuntu:20.04	"/bin/bash"	14 seconds ago	Up 2 seconds		myTestContainer
99a0be2ab835	amazonlinux:latest	"/bin/bash"	21 minutes ago	Up 21 minutes		myAmznLinuxContainer
ade7b6905ca2	ubuntu:20.04	"/bin/bash"	28 minutes ago	Up 28 minutes		objective_brattain
7095aca78d63	ubuntu:18.04	"/bin/bash"	35 minutes ago	Up 35 minutes		nifty_mcclintock

```
[ec2-user@ip-172-31-20-228 ~]$
```

--

## Run Container

- The **docker run** command will do the work of both **docker create** and **docker start** command. This command will create a new container using the specified image.

```
docker run --name=myUbuntuLinuxContainer -i -t ubuntu:20.04 /bin/bash
# enter the `Ctrl + P and Ctrl + Q` escape sequence
docker ps -a
```

--

## Stop Container

- Stopping a running Container means to stop all the processes running in that Container.
- Stopping does not mean killing or ending the process.

```
#Stop the container - docker stop <CONTAINER_ID> OR docker stop <CONTAINER_NAME>
docker ps -a
docker stop myUbuntuLinuxContainer
docker ps -a
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7ebfe07d03bb	ubuntu:20.04	"/bin/bash"	14 seconds ago	Up 13 seconds		myUbuntuLinuxContainer
ceeb928f742f	ubuntu:20.04	"/bin/bash"	7 minutes ago	Up 7 minutes		myTestContainer
99a0be2ab835	amazonlinux:latest	"/bin/bash"	29 minutes ago	Up 29 minutes		myAmznLinuxContainer
ade7b6905ca2	ubuntu:20.04	"/bin/bash"	36 minutes ago	Up 36 minutes		objective_brattain
7095aca78d63	ubuntu:18.04	"/bin/bash"	43 minutes ago	Up 43 minutes		nifty_mcclintock

```
[ec2-user@ip-172-31-20-228 ~]$ docker stop myUbuntuLinuxContainer
myUbuntuLinuxContainer
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7ebfe07d03bb	ubuntu:20.04	"/bin/bash"	About a minute ago	Exited (0) 1 second ago		myUbuntuLinuxContainer
ceeb928f742f	ubuntu:20.04	"/bin/bash"	9 minutes ago	Up 8 minutes		myTestContainer
99a0be2ab835	amazonlinux:latest	"/bin/bash"	30 minutes ago	Up 30 minutes		myAmznLinuxContainer
ade7b6905ca2	ubuntu:20.04	"/bin/bash"	37 minutes ago	Up 37 minutes		objective_brattain
7095aca78d63	ubuntu:18.04	"/bin/bash"	44 minutes ago	Up 44 minutes		nifty_mcclintock

```
[ec2-user@ip-172-31-20-228 ~]$
```

--

## Restart Container

- ```
docker restart myTestContainer
docker ps -a
```

— —

- Removing or deleting the container means destroying all the processes running inside the container and then deleting the Container.
- It's preferred to destroy the container, only if the container is in the **Stopped** state instead of forcefully destroying the running container.

— —



```
[ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
ceeb928f742f   ubuntu:20.04         "/bin/bash"             17 minutes ago Up 3 seconds                myTestContainer
99a0be2ab835   amazonlinux:latest   "/bin/bash"             38 minutes ago Up 38 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04         "/bin/bash"             45 minutes ago Up 45 minutes                objective_brattain
7095aca78d63   ubuntu:18.04         "/bin/bash"             53 minutes ago Up 53 minutes                nifty_mcclintock

[ec2-user@ip-172-31-20-228 ~]$ docker rm myTestContainer
Error response from daemon: You cannot remove a running container ceeb928f742fceebe93ac16c682bf90ca79d95ab5fe88022a19a1136fbd11d8. Stop the container before attempting removal or force remove
[ec2-user@ip-172-31-20-228 ~]$ docker stop myTestContainer
myTestContainer
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
7ebfe07d03bb   ubuntu:20.04         "/bin/bash"             10 minutes ago Exited (0) 8 minutes ago                myUbuntuLinuxContainer
ceeb928f742f   ubuntu:20.04         "/bin/bash"             17 minutes ago Exited (0) 3 seconds ago                myTestContainer
99a0be2ab835   amazonlinux:latest   "/bin/bash"             39 minutes ago Up 39 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04         "/bin/bash"             45 minutes ago Up 45 minutes                objective_brattain
7095aca78d63   ubuntu:18.04         "/bin/bash"             53 minutes ago Up 53 minutes                nifty_mcclintock

[ec2-user@ip-172-31-20-228 ~]$ docker rm myTestContainer
myTestContainer
[ec2-user@ip-172-31-20-228 ~]$ docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
7ebfe07d03bb   ubuntu:20.04         "/bin/bash"             10 minutes ago Exited (0) 8 minutes ago                myUbuntuLinuxContainer
99a0be2ab835   amazonlinux:latest   "/bin/bash"             39 minutes ago Up 39 minutes                myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04         "/bin/bash"             45 minutes ago Up 45 minutes                objective_brattain
7095aca78d63   ubuntu:18.04         "/bin/bash"             53 minutes ago Up 53 minutes                nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$
```

--

- Containers cannot be created with same name

```
docker ps
```

```
docker run --name=myAmznLinuxContainer -i -t amazonlinux:latest /bin/bash
docker: Error response from daemon: Conflict. The container name
"/myAmznLinuxContainer" is already in use by container
"1f2e5e3313be7884d8d2823c68dc58987fe231c56f8a350ab59
d53507e1e96af". You have to remove (or rename) that container to be able to reuse
that name.
```

- docker ps** command attributes
  - CONTAINER ID**: container ID associated with the container.
  - IMAGE**: image from which the Docker container has been created.
  - COMMAND**: command executed during the container launch.
  - CREATED**: Time when the container was created.
  - STATUS**: Current status of the container.
  - PORTS**: if any port has been assigned to the container.
  - NAMES**: The Docker engine auto-generates a random container name. Use **--name** option to specify a customer name in the **docker run** subcommand.

## Changes inside Container Files

- Track the changes inside the containers, connect into shell of the docker and create some directories and files inside the container.

```
docker run -i -t ubuntu:18.04 /bin/bash
cd /home
ls -al
touch file{1..5}
touch {f1.txt,f2.txt,f3.txt}
ls -al
```

pwd

# enter the `Ctrl + P and Ctrl + Q` escape sequence

--

```
[ec2-user@ip-172-31-20-228 ~]$ sudo docker run -i -t ubuntu:18.04 /bin/bash
root@040d0e625bcd:/# cd /home
root@040d0e625bcd:/home# ls -al
total 0
drwxr-xr-x 2 root root 6 Apr 24 2018 .
drwxr-xr-x 1 root root 6 Jan 21 20:46 ..
root@040d0e625bcd:/home# touch file{1..5}
root@040d0e625bcd:/home# touch {f1.txt,f2.txt,f3.txt}
root@040d0e625bcd:/home# ls -al
total 0
drwxr-xr-x 1 root root 113 Jan 21 20:47 .
drwxr-xr-x 1 root root 18 Jan 21 20:46 ..
-rw-r--r-- 1 root root 0 Jan 21 20:47 f1.txt
-rw-r--r-- 1 root root 0 Jan 21 20:47 f2.txt
-rw-r--r-- 1 root root 0 Jan 21 20:47 f3.txt
-rw-r--r-- 1 root root 0 Jan 21 20:47 file1
-rw-r--r-- 1 root root 0 Jan 21 20:47 file2
-rw-r--r-- 1 root root 0 Jan 21 20:47 file3
-rw-r--r-- 1 root root 0 Jan 21 20:47 file4
-rw-r--r-- 1 root root 0 Jan 21 20:47 file5
root@040d0e625bcd:/home# pwd
/home
root@040d0e625bcd:/home#
```

--

- Use `docker diff` command to check the changes made inside the container

```
docker ps
docker diff stoic_gates
```

```
[ec2-user@ip-172-31-20-228 ~]$ docker ps
CONTAINER ID   IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
040d0e625bcd   ubuntu:18.04       "/bin/bash"        About a minute ago   Up About a minute                   stoic_gates
99a0be2ab835   amazonlinux:latest "/bin/bash"        43 minutes ago      Up 43 minutes                   myAmznLinuxContainer
ade7b6905ca2   ubuntu:20.04       "/bin/bash"        49 minutes ago      Up 49 minutes                   objective_brattain
7095aca78d63   ubuntu:18.04       "/bin/bash"        57 minutes ago      Up 57 minutes                   nifty_mcclintock
[ec2-user@ip-172-31-20-228 ~]$ docker diff stoic_gates
C /home
A /home/file3
A /home/file4
A /home/file5
A /home/f1.txt
A /home/f2.txt
A /home/f3.txt
A /home/file1
A /home/file2
[ec2-user@ip-172-31-20-228 ~]$
```