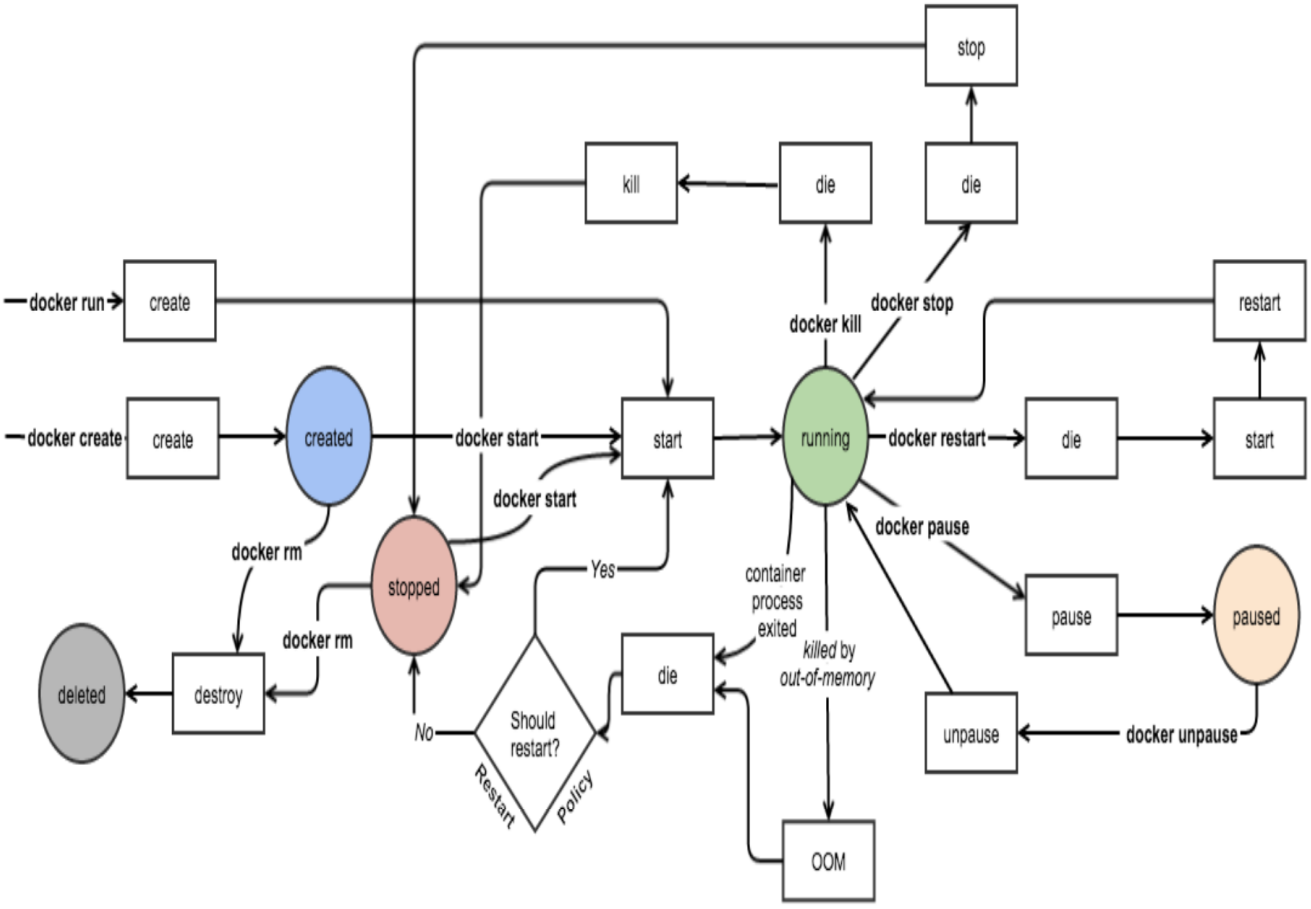
Lab 2 - Managing Docker Containers

Introduction

Docker provides the ability to package and run an application in a loosely isolated environment called a **container**. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight because they don't need the extra load of a hypervisor, but run directly within the host machine's kernel. You can even run Docker containers within host machines that are actually virtual machines!

Docker provides tooling and a platform to **manage the lifecycle of your containers**:

1. Develop your application and its supporting components using containers.
2. The container becomes the unit for distributing and testing your application.
3. When you're ready, deploy your application into your production environment, as a container or an orchestrated service.
4. This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.



1. Managing the Life Cycle of Container

1.1 Login as **“root”** user on **aio110** host:

Copy

```
ssh root@aio110
```

1.2 Creates a new container.

Create a container to run it later on .

Syntax :

```
docker create [OPTIONS] IMAGE [COMMAND] [ARG...]
```

Note: Options are referring as

"**-t**" : tty
"**-i**" : interactive

Copy

```
docker create --name centos-box1 -t -i centos /bin/bash

docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES

ec2fa92a2fe7        centos              "/bin/bash"         6 seconds
ago       Created                                 centos-box1
```

1.3 Run docker container.

Docker runs processes in isolated containers. A container is a process which runs on a host. The host may be local or remote. When an operator executes **docker run**, the container process that runs is isolated in that it has its own file system, its own networking, and its own isolated process tree separate from the host.

The basic docker run command takes this form.

Syntax :

```
docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG….]
```

The docker run command must specify an IMAGE to derive the container from.

Copy

```
docker run --name centos-box2 ubuntu
```

With the docker run [OPTIONS] an operator can add to or override the image defaults set by a developer. And, additionally, operators can override nearly all the defaults set by the Docker runtime itself. The operator's ability to override image.

Copy

```
docker run --name centos-box3 -dit ubuntu
```

Sample Output:

```
a67457b589076e5a645a57641ad2c6529d218bdf6483fd60876c74c88203d647
```

Copy

```
docker ps -a
```

Output:

```
a67457b58907        ubuntu              "/bin/bash"         1 second
ago        Up Less than a second                           centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         14 seconds
ago      Exited (0) 13 seconds ago                        centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         2 minutes
ago       Created                                         centos-box1
```

1.4 Stop container

To stop the container and processes running inside the container by sending **SIGTERM** and then **SIGKILL** after a grace period.

Syntax :

```
docker stop <container-name or container-id>
```

Copy

```
docker stop centos-box3
```

Output:

```
centos-box3
```

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

a67457b58907        ubuntu              "/bin/bash"         2 minutes
ago       Exited (0) Less than a second ago
centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         2 minutes
ago       Exited (0) 2 minutes ago
centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         4 minutes
ago       Created
centos-box1
```

1.5 Start container

Start the container, if present in stopped state

Syntax :

```
docker start <container-name or container-id>
```

Copy

```
docker start centos-box3
```

Output:

```
centos-box3
```

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

a67457b58907        ubuntu              "/bin/bash"         4 minutes
ago       Up Less than a second                           centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         4 minutes
ago       Exited (0) 4 minutes ago                        centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         6 minutes
ago       Created                                         centos-box1
```

1.6 Restart a running container

It is used to restart the container as well as processes running inside the container.

Syntax :

```
docker restart [OPTIONS] CONTAINER [CONTAINER...]
```

Copy

```
docker restart centos-box3
```

Output:

```
centos-box3
```

1.7 Pause container

Used to pause the processes running inside the container.

Syntax :

```
docker pause CONTAINER [CONTAINER...]
```

Copy

```
docker pause centos-box3

docker ps -a
```

Note:A key difference between pausing and stopping containers is in persistence of state. When a container is stopped any resources allocated to it such as memory are released while a paused container does not release its allocated resources.

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

a67457b58907        ubuntu              "/bin/bash"         6 minutes
ago       Up 33 seconds (Paused)                          centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         6 minutes
ago       Exited (0) 6 minutes ago                        centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         8 minutes
ago       Created                                         centos-box1
```

1.8 Unpause container

Unpause all processes within a container.

Syntax :

```
docker unpause CONTAINER [CONTAINER...]
```

Copy

```
docker unpause centos-box3
```

1.9 Daemonized container

Instead of running docker container with an interactive shell it is also possible to let docker container to run as a daemon which means that the docker container would run in the background completely detached from current shell. The following CentOS docker container will start as a daemonized container using "**-d**" option.

Copy

```
docker run --name centos-box4 -d -it centos
```

Sample Output:

```
0b597e0a5c26c9791ffd09def1f60c5485fdf26cf4bcb6495847445b4f624c7e
```

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

0b597e0a5c26        centos              "/bin/bash"         Less than
a second ago   Up Less than a second                       centos-
box4

a67457b58907        ubuntu              "/bin/bash"         7 minutes
ago            Up About a minute                           centos-
box3

93a3410c4d7c        ubuntu              "/bin/bash"         7 minutes
ago            Exited (0) 7 minutes ago                    centos-
box2

ec2fa92a2fe7        centos              "/bin/bash"         9 minutes
ago            Created                                     centos-
box1
```

1.10 Rename the container

Rename an existing container to a NEW_NAME.

Syntax :

```
docker rename OLD_NAME NEW_NAME
```

Copy

```
docker rename centos-box4 newcentos-box4

docker ps -a
```

Output :

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

0b597e0a5c26        centos              "/bin/bash"         2 minutes
ago       Up 2 minutes                                    newcentos-
box4

a67457b58907        ubuntu              "/bin/bash"         9 minutes
ago       Up 3 minutes                                    centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         9 minutes
ago       Exited (0) 9 minutes ago                        centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         11 minutes
ago      Created                                         centos-box1
```

1.11 Docker Images

This command lists the images stored in the local Docker repository.

Syntax :

```
docker images [OPTIONS] [REPOSITORY]
```

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

ubuntu              latest              0458a4468cbc        3 weeks
ago         112MB

centos              latest              ff426288ea90        5 weeks
ago         207MB
```

Copy

```
docker images centos
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

centos              latest              ff426288ea90        5 weeks
ago         207MB
```

1.12 Search the container

Search the Docker Hub for images.

Syntax :

```
docker search [OPTIONS] TERM
```

Copy

```
docker search fedora
```

Copy

```
docker search --filter=stars=3 fedora
```

Note: Options are –

--filter , -f : Filter output based on conditions provided
stars : (int) number of stars the image has

Output:

```
NAME                      DESCRIPTION
STARS     OFFICIAL   AUTOMATED

fedora                    Official Docker builds of Fedora
627       [OK]

mattdm/fedora             A basic Fedora image corresponding
roughly...   49

fedora/apache
34                  [OK]

mattsch/fedora-nzbhydra   Fedora NZBHydra
5                   [OK]
```

1.13 Pull the container

To pull an image or a repository from a registry.

Syntax :

```
docker pull [-a|--all-
tags][help]NAME[:TAG]|[REGISTRY_HOST[:REGISTRY_PORT]/]NAME[:TAG]

-a, --all-tags=true|false
```

Download all tagged images in the repository. The default is false.

Copy

```
docker pull fedora

docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

ubuntu              latest              0458a4468cbc        3 weeks
ago         112MB

centos              latest              ff426288ea90        5 weeks
ago         207MB

fedora              latest              422dc563ca32        3 months
ago         252MB
```

Note: Ask instructor for the difference between dcoker "**run**" and "**pull**"!

1.14 RMI the container

Remove one or more images.

Syntax :

```
docker rmi [OPTIONS] IMAGE [IMAGE...]
```

Copy

```
docker rmi fedora
```

Sample Output:

```
Untagged: fedora:latest

Untagged:
fedora@sha256:25f7dac76b2c88d8b7e0b1d6213d3406e77c7f230bfa1e66bd1cbb81
a944eaaf

Deleted:
sha256:422dc563ca3260ad9ef5c47a1c246f5065d7f177ce51f4dd208efd82967ff18
2
```

```
Deleted:
sha256:d32459d9ce237564fb93573b85cbc707600d43fbe5e46e8eeef22cad914bb51
6
```

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

ubuntu              latest              0458a4468cbc        3 weeks
ago         112MB

centos              latest              ff426288ea90        5 weeks
ago         207MB
```

1.15 Remove the container

Remove one or more containers.

Syntax :

```
docker rm [OPTIONS] CONTAINER [CONTAINER...]
```

Copy

```
docker stop newcentos-box4
```

Copy

```
docker rm newcentos-box4

docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

a67457b58907        ubuntu              "/bin/bash"         18 minutes
ago      Up 12 minutes                                   centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         18 minutes
ago      Exited (0) 18 minutes ago                       centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         20 minutes
ago      Created                                         centos-box1
```

Note: To remove container forcefully use the below command.

```
docker rm newcentos-box4 -f
```

1.16 Save the container

Save one or more images to a tar archive (streamed to **STDOUT** by default).

Syntax :

```
docker save [OPTIONS] IMAGE [IMAGE...]
```

List out an image to keep backup

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

ubuntu              latest              0458a4468cbc        3 weeks
ago         112MB

centos              latest              ff426288ea90        5 weeks
ago         207MB
```

Let’s save the image as **tar** file

Copy

```
docker save ubuntu > ubuntu-backup.tar
```

Copy

```
ls -lh
```

Output:

```
total 111M

-rw-------. 1 root root 6.8K Jul 31  2017 anaconda-ks.cfg

-rwxr-xr-x. 1 root root 1.2K Feb 15 10:41 get_packstack.sh

-rw-------. 1 root root 6.5K Jul 31  2017 original-ks.cfg

-rw-r--r--  1 root root 111M Feb 17 12:30 ubuntu-backup.tar
```

1.17 Load the container

Load an image from a tar archive or STDIN.

Syntax :

```
docker load [OPTIONS]
```

Let’s load an image from a tar file.

Copy

```
docker load --input ubuntu-backup.tar
```

Output:

```
Loaded image: ubuntu:latest
```

1.18 Export the container

Export the contents of a filesystem to a tar archive (streamed to **STDOUT** by default). Export the contents of a container's filesystem using the full or shortened container ID or container name. The output is exported to **STDOUT** and can be redirected to a tar file.

Syntax :

```
docker export [OPTIONS] CONTAINER
```

Copy

```
docker export centos-box3 >centos-box3-latest.tar
```

Copy

```
ls -lh

total 197M

-rw-------. 1 root root 6.8K Jul 31  2017 anaconda-ks.cfg

-rwxr-xr-x. 1 root root 1.2K Feb 15 10:41 get_packstack.sh

-rw-r--r--  1 root root  86M Feb 17 12:38 centos-box3-latest.tar

-rw-------. 1 root root 6.5K Jul 31  2017 original-ks.cfg

-rw-r--r--  1 root root 111M Feb 17 12:30 ubuntu-backup.tar
```

1.19 Import the container

Create an empty filesystem image and import the contents of the tarball (.tar,.tar.gz, .tgz, .bzip, .tar.xz, .txz) into it, then optionally tag it.

Syntax :

```
docker import URL|- [REPOSITORY[:TAG]]
```

Copy

```
docker import centos-box3-latest.tar centos-box3-cenos:ver1
```

Sample Output:

```
sha256:8794fe2a1ab3c4c96bfc12722d99ba38cf9149b0d911c483295f2028a8d297e
6
```

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

centos-box3-cenos         ver1                8794fe2a1ab3        5
seconds ago       85.8MB

ubuntu              latest              0458a4468cbc        3 weeks
ago         112MB

centos              latest              ff426288ea90        5 weeks
ago         207MB
```

1.20 Attach the container

The **docker attach** command allows user to attach to a running container using the container's ID or name, either to view its ongoing output or to control it interactively.

Copy

```
docker run -dit --name test1 centos
```

Output:

```
02454709e9ac2d6feb47aabad33aec3ace09056a0e66df0755e5d693374b8d7b
```

Copy

```
docker attach test1
```

Output:

```
[root@02454709e9ac /]#
```

Verify the list of files located in environment

Copy

```
ls
```

Verify every process on the host:

Copy

```
ps -ef
```

Copy

```
exit
```

Note: On doing an "**exit**" the "***centos***" container was stopped.

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

02454709e9ac        centos              "/bin/bash"         57 seconds
ago      Exited (0) 48 seconds ago                       test1

a67457b58907        ubuntu              "/bin/bash"         33 minutes
ago      Up 27 minutes                                   centos-box3
```

```
93a3410c4d7c        ubuntu              "/bin/bash"         33 minutes
ago      Exited (0) 33 minutes ago                       centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         35 minutes
ago      Created                                         centos-box1
```

Copy

```
docker start test1
```

Copy

```
docker ps -a
```

Create a new name as “**test2**” for the “centos” container

Copy

```
docker run -d --name test2 centos /usr/bin/top -b
```

Copy

```
docker attach test2
```

Note: Press `ctrl+c` to interrupt.

Copy

```
docker ps -a
```

Copy

```
docker start test2
```

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

1cfd342ab793        centos              "/usr/bin/top -b"   24 seconds
ago      Up 4 seconds                                    test2

02454709e9ac        centos              "/bin/bash"         2 minutes
ago       Up 52 seconds                                   test1

a67457b58907        ubuntu              "/bin/bash"         34 minutes
ago      Up 28 minutes                                   centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         35 minutes
ago      Exited (0) 35 minutes ago                       centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         36 minutes
ago      Created                                         centos-box1
```

1.21 Monitoring the container

The docker stats command returns a live data stream for running containers. To limit data to one or more specific containers, specify a list of container names or ids separated by a space. User can specify a stopped container but stopped containers do not return any data.

Note: To exit from each of the **below commands** press `ctrl+c`.

Copy

```
docker stats
```

Copy

```
docker stats -a
```

Copy

```
docker stats test1
```

Copy

```
docker stats test1 test2
```

1.22 Get the Docker Information

Docker-info – Display system-wide information. This command displays system wide information regarding the Docker installation. Information displayed includes the kernel version, number of containers and images. The number of images shown is the number of unique images. The same image tagged under different names is counted only once.

Syntax :

```
docker info
```

Copy

```
docker -D info
```

Output:

```
Containers: 5

 Running: 3

 Paused: 0

 Stopped: 2

Images: 3

Server Version: 17.05.0-ce

....

....

Insecure Registries:

 127.0.0.0/8

Live Restore Enabled: false
```

The global `-D` option tells all docker commands to output debug information.

1.23 Events the container

Get real time events from the server. Get event information from the Docker daemon. Information can include historical information and real-time information.
Docker containers will report the following events: attach, commit, copy, create, destroy, detach, die, exec_create, exec_detach, exec_start, export, kill, oom, pause, rename, resize, restart, start, stop, top, unpause, update.

Syntax :

```
docker events [OPTIONS]
```

Create a **date** variable that will output the current date.

Copy

```
TODAY=$(date +%F)

echo $TODAY
```

Output:

```
2018-02-27
```

Note: %F represents as full date; same as %Y-%m-%d (year-month-date).

Copy

```
docker events --since $TODAY
```

Note: Press `ctrl+c` to exit.

1.24 Inspect the container

Return low-level information on a container or image. This displays all the information available in Docker for a given container or image. By default, this will render all results in a JSON array. If the container and image have the same name, this will return container JSON for unspecified type. If a format is specified, the given template will be executed for each result.

Syntax :

```
docker inspect [OPTIONS] CONTAINER|IMAGE [CONTAINER|IMAGE...]
```

Copy

```
docker inspect --type=image centos
```

1.25 Copy the container

Copy files/folders between a container and the local filesystem. The docker cp utility copies the contents of SRC_PATH to the DEST_PATH. You can copy from the container's file system to the local machine or the reverse, from the local filesystem to the container.

Syntax :

```
docker cp [--help] SRC_PATH CONTAINER:DEST_PATH
```

Copy

```
docker cp ubuntu-backup.tar centos-box3:tmp
```

1.26 Execute the container

Run a command in a running container. The command started using docker exec will only run while the container's primary process (PID 1) is running, and will not be restarted if the container is restarted. If the container is paused, then the docker exec command will wait until the container is unpaused, and then run.

Syntax :

```
docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
```

Copy

```
docker exec -it centos-box3 ls tmp
```

Output:

```
ubuntu-backup.tar
```

1.27 Diff the container

Inspect changes on a container's filesystem. Inspect changes on a container's filesystem. You can use the full or shortened container ID or the container name set using docker run –name option.

Syntax :

```
docker diff [--help] CONTAINER
```

Copy

```
docker diff centos-box3
```

Output

```
C /tmp

A /tmp/ubuntu-backup.tar
```

Note:
C -> Changed
A -> Added

1.28 History the container

Show the history of when and how an image was created.

Syntax :

```
docker history [OPTIONS] IMAGE
```

Copy

```
docker history ubuntu
```

Output

```
IMAGE               CREATED             CREATED BY
SIZE                COMMENT

0458a4468cbc        3 weeks ago         /bin/sh -c #(nop)  CMD
["/bin/bash"]            0B

           3 weeks ago         /bin/sh -c mkdir -p /run/systemd &&
echo '...   7B

           3 weeks ago         /bin/sh -c sed -i
's/^#\s*\(deb.*universe\...   2.76kB

           3 weeks ago         /bin/sh -c rm -rf /var/lib/apt/lists/*
0B

           3 weeks ago         /bin/sh -c set -xe   && echo
'#!/bin/sh' >...   745B

           3 weeks ago         /bin/sh -c #(nop) ADD
file:a3344b835ea6fdc...   112MB
```

1.29 Kill the container

Kill a running container using SIGKILL or a specified signal. The main process inside each container specified will be sent SIGKILL, or any signal specified with option `--signal`.

Syntax :

```
docker kill [OPTIONS] CONTAINER [CONTAINER...]
```

Copy

```
docker kill centos-box3
```

2. Containers and Shells

When you launch a container, you will also use a shell command while launching the container as shown below. This is what we have seen in the earlier chapters when we were working with containers.

2.1 We used this command to create a new container and then used the **Ctrl+P+Q** command to exit out of the container.

It ensures that the container still exists even after we exit from the container.

Copy

```
docker run -it centos /bin/bash
```

Output:

```
[root@866f40e35ed9 /]#
```

Copy

```
ls
```

Exit

Copy

```
exit
```

2.2 We can verify that the container still exists with the Docker **ps** command. If we had to exit out of the container directly, then the container itself would be destroyed.

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                          PORTS               NAMES
```

```
866f40e35ed9        centos              "/bin/bash"         54 seconds
ago      Exited (127) Less than a second ago
clever_williams

314cca64c1ab        centos              "/bin/bash"         5 minutes
ago       Exited (1) 4 minutes ago
admiring_darwin

6d9863aedb5a        centos              "/bin/bash"         6 minutes
ago       Exited (0) 6 minutes ago
serene_wright

54623f43dde9        centos              "/bin/bash"         18 minutes
ago      Exited (127) 16 minutes ago
agitated_carson

c38811b744c0        centos              "/bin/bash"         20 minutes
ago      Exited (127) 19 minutes ago
nostalgic_kilby

23915fc9533b        centos              "/in/bash"          20 minutes
ago      Created
quirky_archimedes

1cfd342ab793        centos              "/usr/bin/top -b"   33 minutes
ago      Up 32 minutes
test2

02454709e9ac        centos              "/bin/bash"         35 minutes
ago      Up 33 minutes
test1

a67457b58907        ubuntu              "/bin/bash"         About an
hour ago   Exited (137) 23 minutes ago
centos-box3

93a3410c4d7c        ubuntu              "/bin/bash"         About an
hour ago   Exited (0) About an hour ago
centos-box2

ec2fa92a2fe7        centos              "/bin/bash"         About an
hour ago   Created
centos-box1
```

3. Cleanup

3.1 To remove all the containers run the below commands:

Copy

```
docker rm `docker ps -a -q` -f
```

Sample Output:

```
89d2c6c2b29e

eb20c240f350
```

3.2 To remove all the images run the below commands:

Copy

```
docker rmi `docker images -q` -f
```

Sample Output:

```
Untagged: centos:latest

Untagged:
centos@sha256:2671f7a3eea36ce43609e9fe7435ade83094291055f1c96d9d1d1d7c
0b986a5d

Deleted:
sha256:ff426288ea903fcf8d91aca97460c613348f7a27195606b45f19ae91776ca23
d

Deleted:
sha256:e15afa4858b655f8a5da4c4a41e05b908229f6fab8543434db79207478511ff
7

Untagged: hello-world:latest

Untagged: hello-
world@sha256:083de497cff944f969d8499ab94f07134c50bcf5e6b9559b27182d3fa
80ce3f7
```

```
Deleted:
sha256:f2a91732366c0332ccd7afd2a5c4ff2b9af81f549370f7a19acd460f87686bc
7

Deleted:
sha256:f999ae22f308fea973e5a25b57699b5daf6b0f1150ac2a5c2ea9d7fecee50fd
f
```

3.3 Verify that containers are removed:

Copy

```
docker ps
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

3.4 Verify that docker images are removed:

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
```