docker top

With this command, you can see the top processes within a container.

Syntax

```
docker top ContainerID
```

Options

- **ContainerID** − This is the Container ID for which you want to see the top processes.

Return Value

The output will show the top-level processes within a container.

Example

```
sudo docker top 9f215ed0b0d3
```

The above command will show the top-level processes within a container.

Output

When we run the above command, it will produce the following result −

docker stop

This command is used to stop a running container.

Syntax

```
docker stop ContainerID
```

Options

- **ContainerID** − This is the Container ID which needs to be stopped.

Return Value

The output will give the ID of the stopped container.

Example

```
sudo docker stop 9f215ed0b0d3
```

The above command will stop the Docker container **9f215ed0b0d3**.

Output

When we run the above command, it will produce the following result −

docker rm

This command is used to delete a container.

Syntax

```
docker rm ContainerID
```

Options

- **ContainerID** − This is the Container ID which needs to be removed.

Return Value

The output will give the ID of the removed container.

Example

```
sudo docker rm 9f215ed0b0d3
```

The above command will remove the Docker container **9f215ed0b0d3**.

Output

When we run the above command, it will produce the following result −

docker stats

This command is used to provide the statistics of a running container.

Syntax

```
docker stats ContainerID
```

Options

- **ContainerID** − This is the Container ID for which the stats need to be provided.

Return Value

The output will show the CPU and Memory utilization of the Container.

Example

```
sudo docker stats 9f215ed0b0d3
```

The above command will provide CPU and memory utilization of the Container **9f215ed0b0d3**.

Output

When we run the above command, it will produce the following result −

```
CONTAINER           CPU %               MEM USAGE/LIMIT     MEM %
NET I/O
07b0b6f434fe        0.00%               416 KiB/1.416 GiB   0.03%
648 B/648 B
```

docker attach

This command is used to attach to a running container.

Syntax

```
docker attach ContainerID
```

Options

- **ContainerID** − This is the Container ID to which you need to attach.

Return Value

None

Example

```
sudo docker attach 07b0b6f434fe
```

The above command will attach to the Docker container **07b0b6f434fe**.

Output

When we run the above command, it will produce the following result −

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
07b0b6f434fe        centos:latest       "/bin/bash"         3 minutes ago
Up 3 minutes                            cocky_pare
demo@ubuntuserver:~$ sudo docker attach 07b0b6f434fe

[root@07b0b6f434fe /]# _
```

Once you have attached to the Docker container, you can run the above command to see the process utilization in that Docker container.

```
top - 15:24:06 up  2:06,  0 users,  load average: 0.00, 0.01, 0.02
Tasks:   2 total,   1 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
KiB Mem :  1484856 total,  1057152 free,    52368 used,   375336 buff/cache
KiB Swap:  1519612 total,  1519612 free,        0 used.  1403868 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   11784   2992   2644 S   0.0  0.2   0:00.01 bash
   15 root      20   0   51864   3772   3272 R   0.0  0.3   0:00.00 top
```

docker pause

This command is used to pause the processes in a running container.

Syntax

```
docker pause ContainerID
```

Options

- **ContainerID** − This is the Container ID to which you need to pause the processes in the container.

Return Value

The ContainerID of the paused container.

Example

```
sudo docker pause 07b0b6f434fe
```

The above command will pause the processes in a running container **07b0b6f434fe**.

Output

When we run the above command, it will produce the following result −

```
demo@ubuntuserver:~$ sudo docker ps
[sudo] password for demo:
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
07b0b6f434fe        centos:latest       "/bin/bash"         18 minutes ago
Up 18 minutes                           cocky_pare
demo@ubuntuserver:~$ sudo docker pause 07b0b6f434fe
07b0b6f434fe
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                  PORTS               NAMES
07b0b6f434fe        centos:latest       "/bin/bash"         19 minutes ago
Up 19 minutes (Paused)                      cocky_pare
demo@ubuntuserver:~$ _
```

docker unpause

This command is used to **unpause** the processes in a running container.

Syntax

```
docker unpause ContainerID
```

Options

- **ContainerID** − This is the Container ID to which you need to unpause the processes in the container.

Return Value

The ContainerID of the running container.

Example

```
sudo docker unpause 07b0b6f434fe
```

The above command will unpause the processes in a running container: 07b0b6f434fe

Output

When we run the above command, it will produce the following result −

```
demo@ubuntuserver:~$ sudo docker unpause 07b0b6f434fe
07b0b6f434fe
demo@ubuntuserver:~$
```

docker kill

This command is used to kill the processes in a running container.

Syntax

```
docker kill ContainerID
```

Options

- **ContainerID** − This is the Container ID to which you need to kill the processes in the container.

Return Value

The ContainerID of the running container.

Example

```
sudo docker kill 07b0b6f434fe
```

The above command will kill the processes in the running container **07b0b6f434fe**.

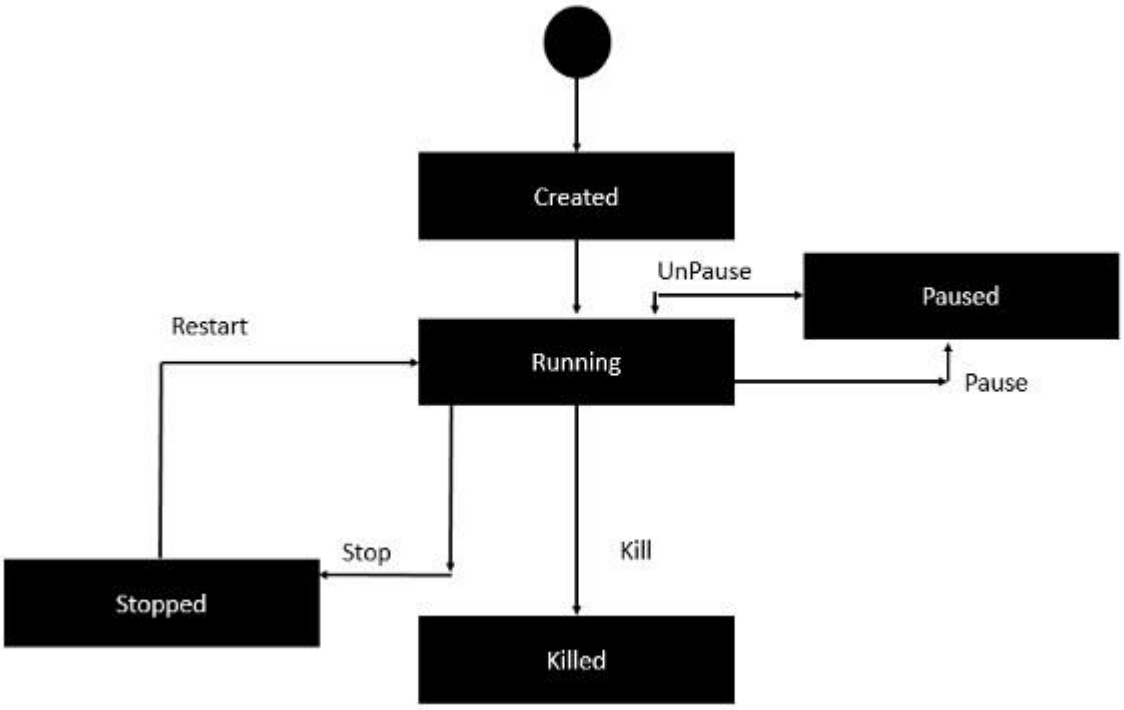
Output

When we run the above command, it will produce the following result −

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
07b0b6f434fe        centos:latest       "/bin/bash"         23 minutes ago
Up 23 minutes                           cocky_pare
demo@ubuntuserver:~$ sudo docker kill 07b0b6f434fe
07b0b6f434fe
demo@ubuntuserver:~$
```

Docker – Container Lifecycle

The following illustration explains the entire lifecycle of a Docker container.



- Initially, the Docker container will be in the **created** state.
- Then the Docker container goes into the running state when the Docker **run** command is used.
- The Docker **kill** command is used to kill an existing Docker container.
- The Docker **pause** command is used to pause an existing Docker container.
- The Docker **stop** command is used to pause an existing Docker container.
- The Docker **run** command is used to put a container back from a **stopped** state to a **running** state.