**Docker Containers**

Containers are instances of Docker images that can be run using the Docker **run** command. The basic purpose of Docker is to run containers. Let’s discuss how to work with containers.

**Running a Container**

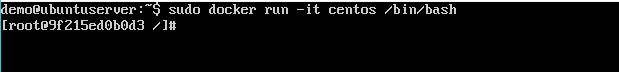


Running of containers is managed with the Docker **run** command. To run a container in an interactive mode, first launch the Docker container.



sudo docker run –it centos /bin/bash

Then hit Crtl+p and you will return to your OS shell.



You will then be running in the instance of the CentOS system on the Ubuntu server.

**Listing of Containers**



One can list all of the containers on the machine via the **docker ps** command. This command is used to return the currently running containers.

docker ps

**Syntax**

docker ps

**Options**

None

**Return Value**

The output will show the currently running containers.

**Example**

sudo docker ps

**Output**

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



Let’s see some more variations of the **docker ps** command.

**docker ps -a**

This command is used to list all of the containers on the system.

**Syntax**

docker ps -a

**Options**

* **─a** –It tells the **docker ps** command to list all of the containers on the system.

**Return Value**

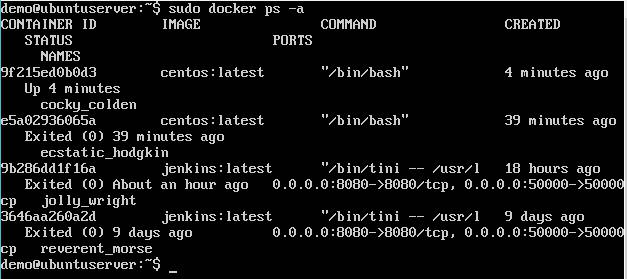
The output will show all containers.

**Example**

sudo docker ps -a

**Output**

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



**docker history**



With this command, you can see all the commands that were run with an image via a container.

**Syntax**

docker history ImageID

**Options**

* **ImageID** –This is the Image ID for which you want to see all the commands thatwere run against it.

**Return Value**

The output will show all the commands run against that image.

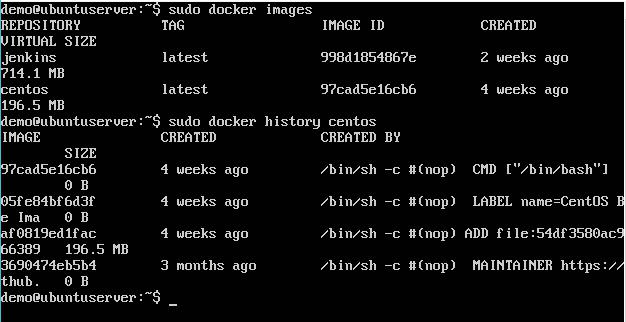
**Example**

sudo docker history centos

The above command will show all the commands that were run against the **centos** image.

**Output**

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



**Working with Docker container**

**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.

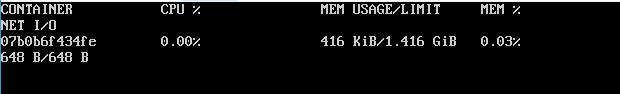
**Example**

sudo docker rm 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:



**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**

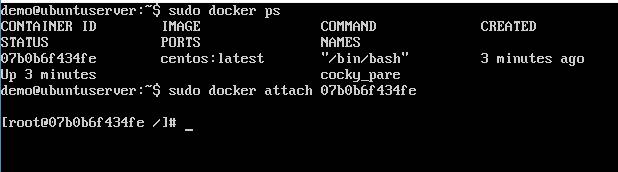


sudo docker attach 07b0b6f434fe

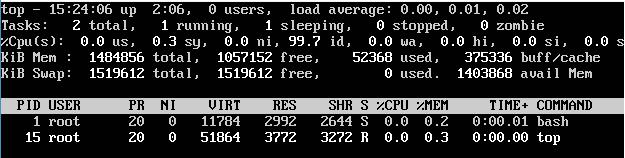
The above command will attach to the Docker container **07b0b6f434fe**.

**Output**

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



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



**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 processesin 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:



**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 theprocesses in the container.

**Return Value**

The ContainerID of the running container.

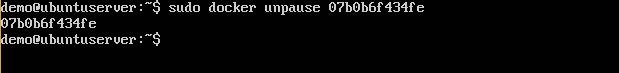
**Example**

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:



**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 inthe container.

**Return Value**

The ContainerID of the running container.

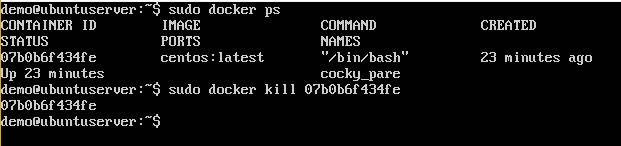
**Example**

sudo docker kill 07b0b6f434fe

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

**Output**

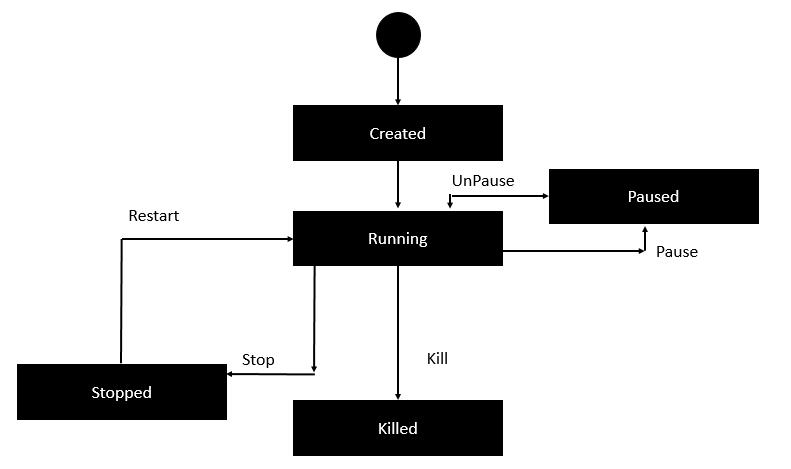
When we run the above command, it will produce the following result:



**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.