**Docker Images**

In Docker, everything is based on Images. An image is a combination of a file system and parameters. Let’s take an example of the following command in Docker.

docker run hello-world

* The Docker command is specific and tells the Docker program on the Operating System that something needs to be done.
* The **run** command is used to mention that we want to create an instance of an image, which is then called a **container**.
* Finally, "hello-world" represents the image from which the container is made.

Now let’s look at how we can use the CentOS image available in Docker Hub to run CentOS on our Ubuntu machine. We can do this by executing the following command on our Ubuntu machine:

sudo docker run centos –it /bin/bash

Note the following points about the above **sudo** command:

* We are using the **sudo** command to ensure that it runs with **root** access.
* Here, **centos** is the name of the image we want to download from Docker Hub and install on our Ubuntu machine.
* **─it** is used to mention that we want to run in **interactive mode**.
* **/bin/bash** is used to run the bash shell once CentOS is up and running.

**Displaying Docker Images**

To see the list of Docker images on the system, you can issue the following command.

docker images

This command is used to display all the images currently installed on the system.

**Syntax**

docker images

**Options**

None

**Return Value**

The output will provide the list of images on the system.

**Example**



sudo docker images

**Output**

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



From the above output, you can see that the server has three images: **centos**, **newcentos**, and **jenkins**. Each image has the following attributes:

* **TAG** –This is used to logically tag images.
* **Image ID** –This is used to uniquely identify the image.
* **Created** –The number of days since the image was created.
* **Virtual Size** –The size of the image.

**Downloading Docker Images**

Images can be downloaded from Docker Hub using the Docker **run** command. Let’s see in detail how we can do this.

**Syntax**

The following syntax is used to run a command in a Docker container.

docker run image

**Options**

* **Image** –This is the name of the image which is used to run the container.

**Return Value**

The output will run the command in the desired container.

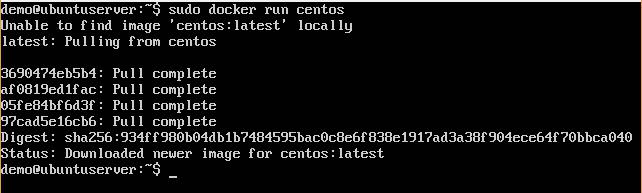
**Example**

sudo docker run centos

This command will download the **centos** image, if it is not already present, and run the OS as a container.

**Output**

When we run the above command, we will get the following result:



You will now see the CentOS Docker image downloaded. Now, if we run the Docker **images** command to see the list of images on the system, we should be able to see the **centos** image as well.



**Removing Docker Images**

The Docker images on the system can be removed via the **docker rmi** command. Let’s look at this command in more detail.

docker rmi

This command is used to remove Docker images.

**Syntax**

docker rmi ImageID

**Options**

* **ImageID** –This is the ID of the image which needs to be removed.

**Return Value**

The output will provide the Image ID of the deleted Image.

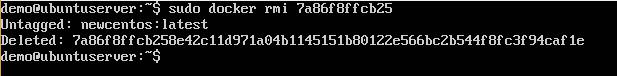
**Example**

sudo docker rmi 7a86f8ffcb25

Here, **7a86f8ffcb25** is the Image ID of the **newcentos** image.

**Output**

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



Let’s see some more Docker commands on images.

**docker images -q**

This command is used to return only the Image ID’s of the images.

**Syntax**

docker images -q

**Options**

* **q** –It tells the Docker command to return the Image ID’sonly.

**Return Value**

The output will show only the Image ID’s of the images on the Docker host.

**Example**

sudo docker images -q

**Output**

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

**docker inspect**

This command is used see the details of an image or container.

**Syntax**

docker inspect Repository

**Options**

* **Repository** –This is the name of the Image.

**Return Value**

The output will show detailed information on the Image.

**Example**

sudo docker inspect jenkins

**Output**

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

