**Docker**

**What is docker?**

Docker is a container management service. The keywords of Docker are **develop, ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

The initial release of Docker was in March 2013 and since then, it has become the buzzword for modern world development, especially in the face of Agile-based projects.

**Features of Docker**

* Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
* With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
* You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
* Since Docker containers are pretty lightweight, they are very easily scalable.

**Components of Docker**

Docker has the following components

* **Docker for Mac** − It allows one to run Docker containers on the Mac OS.
* **Docker for Linux** − It allows one to run Docker containers on the Linux OS.
* **Docker for Windows** − It allows one to run Docker containers on the Windows OS.
* **Docker Engine** − It is used for building Docker images and creating Docker containers.
* **Docker Hub** − This is the registry which is used to host various Docker images.
* **Docker Compose** − This is used to define applications using multiple Docker containers.

**Docker HUB:**

Docker Hub is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub. In this chapter, we will see how to download and the use the Jenkins Docker image from Docker hub.

The official site for Docker hub is − <https://www.docker.com/community-edition#/add_ons>

**Step 1** − First you need to do a simple sign-up on Docker hub.

**Step 2** − Once you have signed up, you will be logged into Docker Hub.

**Step 3** − Next, let’s browse and find the Jenkins image.

**Step 4** − If you scroll down on the same page, you can see the Docker **pull** command. This will be used to download the Jenkins image onto the local Ubuntu server.

**Step 5** − Now, go to the Ubuntu server and run the following command −

sudo docker pull jenkins

To run Jenkins, you need to run the following command −

sudo docker run -p 8080:8080 -p 50000:50000 jenkins

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

* We are using the **sudo** command to ensure it runs with root access.
* Here, **jenkins** is the name of the image we want to download from Docker hub and install on our Ubuntu machine.
* **-p** is used to map the port number of the internal Docker image to our main Ubuntu server so that we can access the container accordingly.

**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 -it centos /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.

### **Example**

sudo docker images



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.

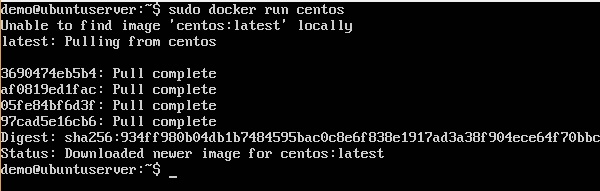
### **Example**

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 −

