Docker Installation & Configuration

October 19, 2019 By Raj Chandel

Docker services are extensively used in IT operations, so it is very important that you start learning from docker basics. In this article, we will cover the installation and setup of the docker, along with its specific uses.

Learn web application in

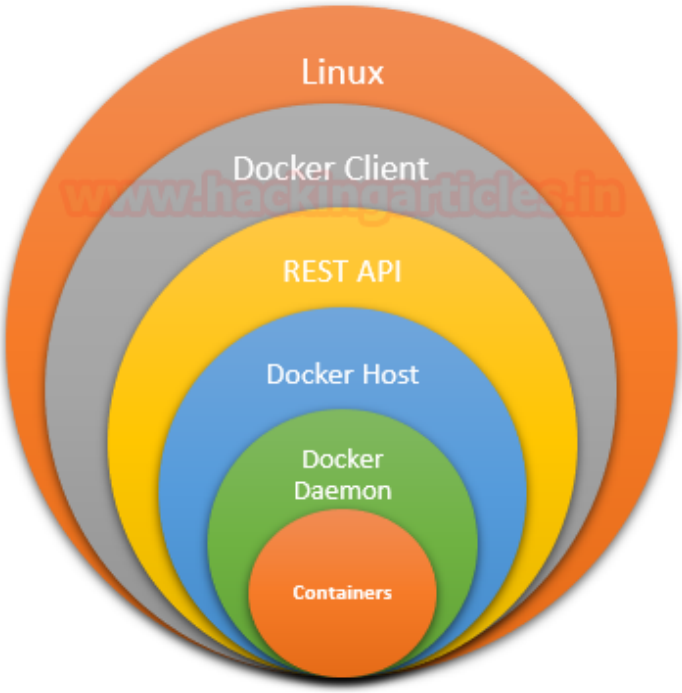
Table of Content

Introduction to Docker

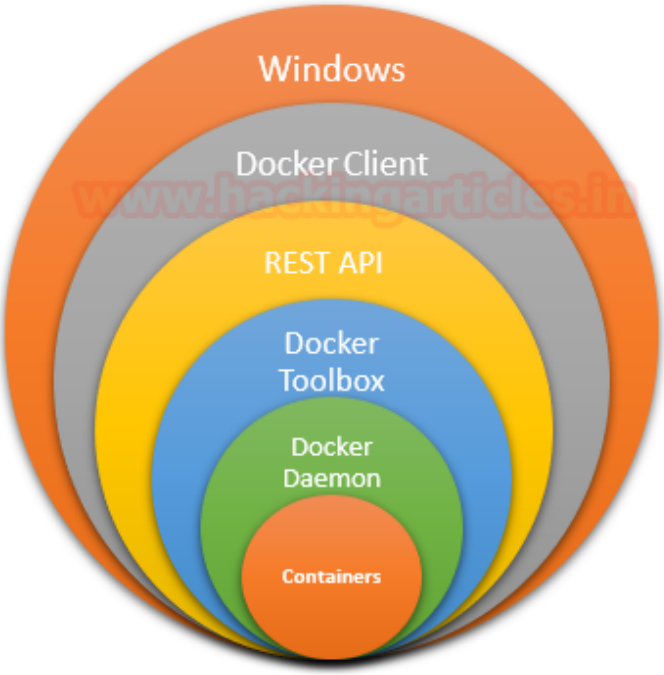
Docker is a third-party tool developed to create an isolated environment to execute any application. These applications are run using containers. These containers are unique because they bring together all the dependencies of an application into a single package and deploy it.

Now, to work with docker you will need to install docker-engine in your host. It is a foundation to the docker system, which basically runs as a client-server application. Its daemon process is referred to as server and the command-line interface is referred to as a client and REST API is used to create a communication link between client and server.

In Linux, docker client interacts with docker server through the CLI. Here, the terminal is docker client and docker host will run the docker daemon.



Whereas in windows, to work with docker, we need to install docker toolbox component in docker host in order to set up an environment on your Windows or iOS.



Docker and its terminology

When working with docker, one should be familiar with the following terms :

- **Docker Hub:** It is a repository which available to all who uses docker through cloud. Through docker hub, one can create, store, test, pull and share container images.
- **Docker Images :** Docker image acts as a template in order to create container. Build command is used to create docker images. Docker images makes it easy.

- **Docker containers :** Containers are said to be isolated environment provided to the docker image and its dependencies so that it can run independently. The focus of deploying a container is to update or repair an application or just simply modify it and share it. When working on an image, container lets you create a layer of a single command used which make it easy to modify it, or upgrade or degrade is version.
- **Docker Registry :** All the docker images are stored in docker registry. User can either can have local registry on their system or they can have a public one like docker hub.

Advantages of docker

- Easy to use
- Faster scaling systems
- Better software delivery
- Flexibility
- Provides isolated environment
- Supports software-defined networking
- Rapid deployment
- Security

Installation and usage

To install docker, simply open the terminal of Linux and type the following command :

```
apt install docker.io
```

```
root@ubuntu:~# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no
  linux-headers-4.15.0-29 linux-headers-4.15.0-29-generic
  linux-image-4.15.0-29-generic linux-modules-4.15.0-29-generi
  linux-modules-extra-4.15.0-29-generic
Use 'apt autoremove' to remove them.
Suggested packages:
  aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse
The following NEW packages will be installed:
  docker.io
0 upgraded, 1 newly installed, 0 to remove and 61 not upgraded
Need to get 0 B/30.7 MB of archives.
```

To check the version one can use the following command :

```
docker --version
```

Further, you can run help command in docker, which is as follows, to know all the options that docker provides at your service.

```
docker --help
```

```
root@ubuntu:~# docker --version
Docker version 18.09.7, build 2d0083d
root@ubuntu:~# docker --help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default "/home
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level ("debug"|"info"|"warn"|"e
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default "/h
      --tlscert string     Path to TLS certificate file (default "/home/ra
      --tlskey string      Path to TLS key file (default "/home/raj/.docke
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  builder     Manage builds
  config      Manage Docker configs
  container   Manage containers
  engine      Manage the docker engine
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  trust       Manage trust on Docker images
  volume      Manage volumes
```

Once the docker is up and running, you can run or pull any image in your docker container. For instance, here we have run hello-world. When you run the following command, it will first check your local repository; if the image is not available there then it will pull it from docker hub.

```
docker run hello-world
```

```
root@ubuntu:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:451ce787d12369c5df2a32c85e5a03d52cbcef6eb3586dd03075f3034f10adcc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

As we have explained before, CLI works as a client, so directly from the terminal, you can search for any image you like. Like, here we have searched for ubuntu. One thing to remember here is that image with more stars will be the most authentic one.

```
docker search ubuntu
```

Once you find your image, you can pull it into your container with the following command :

```
docker pull ubuntu
```

```
root@ubuntu:~# docker search ubuntu
NAME                                                    DESCRIPTION                                     STARS
ubuntu                                                  Ubuntu is a Debian-based Linux operating sys…   9855
dorowu/ubuntu-desktop-lxde-vnc                          Docker image to provide HTML5 VNC interface …   334
rastasheep/ubuntu-sshd                                  Dockerized SSH service, built on top of offi…   228
consol/ubuntu-xfce-vnc                                  Ubuntu container with "headless" VNC session…   186
ubuntu-upstart                                          Upstart is an event-based replacement for th…   99
ansible/ubuntu14.04-ansible                             Ubuntu 14.04 LTS with ansible                   97
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5    ubuntu-16-nginx-php-phpmyadmin-mysql-5          50
ubuntu-debootstrap                                      debootstrap --variant=minbase --components=m…   40
nuagebec/ubuntu                                         Simple always updated Ubuntu docker images w…   24
i386/ubuntu                                             Ubuntu is a Debian-based Linux operating sys…   18
1and1internet/ubuntu-16-apache-php-5.6                  ubuntu-16-apache-php-5.6                        14
ppc64le/ubuntu                                          Ubuntu is a Debian-based Linux operating sys…   13
1and1internet/ubuntu-16-apache-php-7.0                  ubuntu-16-apache-php-7.0                        13
eclipse/ubuntu_jdk8                                     Ubuntu, JDK8, Maven 3, git, curl, nmap, mc, …   11
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10 ubuntu-16-nginx-php-phpmyadmin-mariadb-10      11
1and1internet/ubuntu-16-nginx-php-5.6                   ubuntu-16-nginx-php-5.6                         8
1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4       ubuntu-16-nginx-php-5.6-wordpress-4             7
1and1internet/ubuntu-16-apache-php-7.1                  ubuntu-16-apache-php-7.1                        6
darksheer/ubuntu                                        Base Ubuntu Image -- Updated hourly             5
1and1internet/ubuntu-16-nginx-php-7.0                   ubuntu-16-nginx-php-7.0                         4
pivotaldata/ubuntu                                      A quick freshening-up of the base Ubuntu doc…   2
1and1internet/ubuntu-16-sshd                            ubuntu-16-sshd                                  1
1and1internet/ubuntu-16-php-7.1                         ubuntu-16-php-7.1                               1
smartentry/ubuntu                                       ubuntu with smartentry                          1
pivotaldata/ubuntu-gpdb-dev                             Ubuntu images for GPDB development              0
root@ubuntu:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
35c102085707: Pull complete
251f5509d51d: Pull complete
8e829fe70a46: Pull complete
6001e1789921: Pull complete
Digest: sha256:d1d454df0f579c6be4d8161d227462d69e163a8ff9d20a847533989cf0c94d90
Status: Downloaded newer image for ubuntu:latest
```

Now to check how many images you have in your docker, simply type the following command :

```
docker images
```

```
root@ubuntu:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              a2a15febcdf3        7 days ago          64.2MB
busybox             latest              db8ee88ad75f        4 weeks ago         1.22MB
hello-world         latest              fce289e99eb9        7 months ago        1.84kB
root@ubuntu:~#
```

To remove any image, use the following command :

```
docker rmi hello-world
```

Here, rmi refers to remove image.

```
root@ubuntu:~# docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:451ce787d12369c5df2a32c85e5a03d52cbcef6eb3586d
Deleted: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675
Deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c5483994
root@ubuntu:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
ubuntu              latest              a2a15febcdf3        7 days ago
busybox             latest              db8ee88ad75f        4 weeks ago
```

Now, in the details given by ps command, you can see that the name of our ubuntu image is **adoring curie**, which is a random name generated by docker for every image. To, rename this name we can use the following command :

```
docker run -it -d ubuntu
docker run -it -d --name "ignite" ubuntu
docker ps
```

And you can confirm with the ps command again that the name has been changed as shown in the image below :

```
root@ubuntu:~# docker run -it -d ubuntu
f490699a0e797ee982da09564aa429892b238e682b51f583b9997a54560120dc
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
f490699a0e79        ubuntu              "/bin/bash"         7 seconds ago       Up 5 seconds                            adoring_curie
root@ubuntu:~# docker run -it -d --name "ignite" ubuntu
bea2790843e9af2ff0d183add69dd494090e890f0ab065b7838b3a0c3b4fa574
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
bea2790843e9        ubuntu              "/bin/bash"         4 seconds ago       Up 2 seconds                            ignite
f490699a0e79        ubuntu              "/bin/bash"         34 seconds ago      Up 32 seconds                           adoring_curie
```

The docker attaches command permits you to attach to a running container using the container ID or name, you can use one instance of shell only though attach command. But if you crave to open new terminal with new instance of container's shell, we just need run docker exec.

```
docker attach ignite
docker exec -i -ignite /bin/bash
```

```
root@ubuntu:~# docker attach ignite
root@bea2790843e9:/# id
uid=0(root) gid=0(root) groups=0(root)
root@bea2790843e9:/# whoami
root
root@bea2790843e9:/# uname -a
Linux bea2790843e9 4.15.0-58-generic #64-Ubuntu SMP Tue
```

Using the ps command we can see all the processes that are running in docker. There, for this, type :

```
docker ps
docker ps -a
```

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
bea2790843e9        ubuntu              "/bin/bash"         About a minute ago
f490699a0e79        ubuntu              "/bin/bash"         About a minute ago
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
bea2790843e9        ubuntu              "/bin/bash"         About a minute ago
f490699a0e79        ubuntu              "/bin/bash"         About a minute ago
33766821263e        ubuntu              "--name ignite"     3 minutes ago
root@ubuntu:~#
```

To stop the running container, you can use stop command as shown in the below image, we have stopped the container and its process which can be confirm with the help of process command. As result there should be no running process for ignite.

```
docker stop <docker-container >
docker rm ignite
docker ps -a
```

```
root@ubuntu:~# docker stop ignite
ignite
root@ubuntu:~# docker rm ignite
ignite
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
f490699a0e79        ubuntu              "/bin/bash"         4 minutes ago
33766821263e        ubuntu              "--name ignite"     5 minutes ago
root@ubuntu:~#
```

If you can export the docker filesystem as a archive, use export command to compress the filesystem of a docker container into tar. The export commands fetch the whole container like a snapshot of a regular VM.

```
docker export <container ID> | gzip > {path for tar} filename.gz
```

```
root@ubuntu:~# docker export f490699a0e79 | gzip > ignitelab.gz
root@ubuntu:~# ls
Desktop  Documents  Downloads  examples.desktop  icmp.pcap  icmp.pdml  icmp.xml  ignitelab.gz  Music
root@ubuntu:~#
```

```
docker export <container name> | gzip > {path for tar} filename.tar
```

It will give you a flat .tar archive containing the filesystem of your container.

```
oot@ubuntu:~# docker ps
ONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
001c000dc22        ubuntu              "/bin/bash"         4 minutes ago       Up 4 minutes                            ignite
oot@ubuntu:~# docker export ignite | gzip >/home/raj/docker/ignitelab.gz
oot@ubuntu:~# cd docker/
oot@ubuntu:~/docker# ls
gnitelab.gz
```

When you will export container as tar file, the file has hash value which can read as:

```
cat {path of exported tar file} |docker import – newignitelab
```

```
root@ubuntu:~# cat /home/raj/docker/ignitelab.gz | docker import - newignitelab
sha256:1b1d9b019d285a2d08a354520d2aa3b6f33f48f0b445ecacea85eff29c8fa939
root@ubuntu:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
newignitelab        latest              1b1d9b019d28        4 seconds ago       64.2MB
ubuntu              latest              a2a15febcdf3        7 days ago          64.2MB
busybox             latest              db8ee88ad75f        4 weeks ago         1.22MB
```

In order to save the image of container which you can upload on other docker use save command. You can subsequently load this “saved” images into a new docker instance and create containers from these images.

```
docker save <container name> | gzip > {path for tar} filename.tar
docker load -i /home/raj/docker/igniteimage.tar
```

```
root@ubuntu:~# docker save newignitelab > /home/raj/docker/igniteimage.tar
root@ubuntu:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              a2a15febcdf3        7 days ago          64.2MB
busybox             latest              db8ee88ad75f        4 weeks ago         1.22MB
root@ubuntu:~# docker load -i /home/raj/docker/igniteimage.tar
c24f1d32a59a: Loading layer [==================================================>]  66.56MB/66.56MB
Loaded image: newignitelab:latest
root@ubuntu:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
newignitelab        latest              1b1d9b019d28        4 minutes ago       64.2MB
ubuntu              latest              a2a15febcdf3        7 days ago          64.2MB
busybox             latest              db8ee88ad75f        4 weeks ago         1.22MB
```

In order to clear all image and or stop all process of the container. It will pack the layers and metadata of all the chain required to build the image.

```
docker rm -f $(docker ps -aq)
```

```
root@ubuntu:~# docker rm -f  $(docker ps -aq)
d2c68b69da8e
33766821263e
```

To learn how to set up vulnerable web application setup using docker from **here**.

Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact ~~**here**~~