

Install Docker & Container Manipulation

Step 1: Install and Configure Docker

1. Docker binaries are incorporated into RHEL/CentOS 7 extras repositories, the installation process being pretty simple. Install Docker package by issuing the following command with root privileges:

Install Docker on RHEL and CentOS 7

```
# yum install docker
```

```
Loading mirror speeds from cached hostfile
* base: centos.excellmedia.net
* epel: mirrors.hustunique.com
* extras: centos.excellmedia.net
* updates: centos.excellmedia.net
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:1.8.2-10.el7.centos will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version           Repository        Size
=====
Installing:
docker                 x86_64           1.8.2-10.el7.centos  extras            10 M

Transaction Summary
=====
Install 1 Package

Total download size: 10 M
Installed size: 46 M
Is this ok [y/d/N]: _
```

Install Docker on CentOS and RHEL 7

Install Docker on RHEL and CentOS 6

To install Docker, the [Epel repositories](#) must be enabled on your system by issuing the following command:

```
# yum install epel-release
```

```
# yum install docker-io
```

```

---> Package luafilesystem.x86_64 0:1.4.2-1.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version           Repository        Size
=====
Installing:
docker-io              x86_64           1.7.1-2.el6       epel               4.6 M
Installing for dependencies:
libcgroup              x86_64           0.40.rc1-16.el6   base              129 k
lua-alt-getopt          noarch           0.7.0-1.el6       epel               6.9 k
lua-filesystem         x86_64           1.4.2-1.el6       epel               24 k
lua-lxc                x86_64           1.0.8-1.el6       epel               16 k
lxc                    x86_64           1.0.8-1.el6       epel              122 k
lxc-libs               x86_64           1.0.8-1.el6       epel              255 k

Transaction Summary
=====
Install      7 Package(s)

Total download size: 5.1 M
Installed size: 20 M
Is this ok [y/N]: y_

```

Install Docker on RHEL and CentOS 6

2. After, Docker package has been installed, start the daemon, check its status and enable it system wide using the below commands:

On RHEL/CentOS 7

```

# systemctl start docker

# systemctl status docker

# systemctl enable docker

```

```

[root@tecmin ~]# systemctl start docker.service
[root@tecmin ~]# systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2016-01-27 02:19:12 EST; 9s ago
     Docs: http://docs.docker.com
    Main PID: 3980 (docker)
    CGroup: /system.slice/docker.service
            └─3980 /usr/bin/docker daemon --selinux-enabled

Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.587277045-0...e"
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.587329805-0...e"
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.589896374-0...1"
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.593921596-0...e"
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.802175913-0...."
Jan 27 02:19:12 tecmint docker[3980]: ..
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.825428906-0...."
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.825477738-0...n"
Jan 27 02:19:12 tecmint docker[3980]: time="2016-01-27T02:19:12.825515148-0...os
Jan 27 02:19:12 tecmint systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[root@tecmin ~]# systemctl enable docker.service_

```

Enable Docker on RHEL and CentOS 7

On RHEL/CentOS 6

```

# service docker start

# service docker status

# chkconfig docker on

```

```
[root@tecmin ~]# service docker start
Starting docker: [ OK ]
[root@tecmin ~]# service docker status
docker (pid 3798) is running...
[root@tecmin ~]# chkconfig docker on
[root@tecmin ~]# _
```

Enable Docker on RHEL and CentOS 6

3. Finally, run a container test image to verify if Docker works properly, by issuing the following command:

```
# docker run hello-world
```

If you can see the below message, then everything is in the right place.

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

```
975b84d108f1: Pull complete
Digest: sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc1ca36966a7
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.
 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 Hub account:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/userguide/

[root@tecmin ~]# _
```

Docker Hello World

4. Now, you can run a few basic Docker commands to get some info about Docker:

For system-wide information on Docker

```
# docker info
```

```
Pool Name: docker-8:3-525701-pool
Pool Blocksize: 65.54 kB
Backing Filesystem: extfs
Data file: /dev/loop0
Metadata file: /dev/loop1
Data Space Used: 307.7 MB
Data Space Total: 107.4 GB
Data Space Available: 36.47 GB
Metadata Space Used: 827.4 kB
Metadata Space Total: 2.147 GB
Metadata Space Available: 2.147 GB
Udev Sync Supported: true
Deferred Removal Enabled: false
Data loop file: /var/lib/docker/devicemapper/devicemapper/data
Metadata loop file: /var/lib/docker/devicemapper/devicemapper/metadata
Library Version: 1.02.95-RHEL6 (2015-09-08)
Execution Driver: native-0.2
Logging Driver: json-file
Kernel Version: 2.6.32-573.12.1.el6.x86_64
Operating System: <unknown>
CPUs: 1
Total Memory: 996.3 MiB
Name: tecmint
ID: 7Z4N:IIEA:RNIX:QUGR:4LV4:JFDG:FEYP:YKE6:X2DC:350L:B7QI:U3KT
[root@tecmint ~]# _
```

Check Docker Info

For Docker version

```
# docker version
```

```
[root@tecmint ~]# docker version
Client version: 1.7.1
Client API version: 1.19
Go version (client): go1.4.2
Git commit (client): 786b29d/1.7.1
OS/Arch (client): linux/amd64
Server version: 1.7.1
Server API version: 1.19
Go version (server): go1.4.2
Git commit (server): 786b29d/1.7.1
OS/Arch (server): linux/amd64
[root@tecmint ~]# _
```

Check Docker Version

5. To get a list of all available Docker commands type docker on your console.

```
# docker
```

```
Usage: docker [OPTIONS] COMMAND [arg...]

A self-sufficient runtime for linux containers.

Options:
  --api-cors-header=      Set CORS headers in the remote API
  -b, --bridge=           Attach containers to a network bridge
  --bip=                 Specify network bridge IP
  -D, --debug=false       Enable debug mode
  -d, --daemon=false      Enable daemon mode
  --default-gateway=      Container default gateway IPv4 address
  --default-gateway-v6=   Container default gateway IPv6 address
  --default-ulimit=[]     Set default ulimits for containers
  --dns=[]               DNS server to use
  --dns-search=[]        DNS search domains to use
  -e, --exec-driver=native Exec driver to use
  --exec-opt=[]          Set exec driver options
  --exec-root=/var/run/docker Root of the Docker execdriver
  --fixed-cidr=           IPv4 subnet for fixed IPs
  --fixed-cidr-v6=        IPv6 subnet for fixed IPs
  -G, --group=docker      Group for the unix socket
  -g, --graph=/var/lib/docker Root of the Docker runtime
  -H, --host=[]           Daemon socket(s) to connect to
  :
```

List Docker Commands

Step 2: Download a Docker Image

6. In order to start and run a Docker container, first an image must be downloaded from [Docker Hub](#) on your host. Docker Hub offers a great deal of free images from its repositories. To search for a Docker image, Ubuntu for instance, issue the following command:

```
# docker search ubuntu
```

```
[root@vbox ~]# docker search ubuntu
INDEX      NAME                                DESCRIPTION                                STARS    OFFICIAL    AUTOMATED
docker.io  docker.io/ubuntu                   Ubuntu is a Debian-based Linux operating s... 3040     [OK]
docker.io  docker.io/ubuntu-upstart           Upstart is an event-based replacement for ... 60       [OK]
docker.io  docker.io/torusware/speedus-ubuntu Always updated official Ubuntu docker imag... 25
docker.io  docker.io/ubuntu-debootstrap       debootstrap --variant=minbase --components... 23       [OK]
docker.io  docker.io/tleyden5iwx/ubuntu-cuda  Ubuntu 14.04 with CUDA drivers pre-installed 19
docker.io  docker.io/rastasheep/ubuntu-sshd   Dockerized SSH service, built on top of of... 18       [OK]
docker.io  docker.io/neurodebian              NeuroDebian provides neuroscience research... 15       [OK]
docker.io  docker.io/sameersbn/ubuntu         docker.io/sameersbn/ubuntu                  5        [OK]
docker.io  docker.io/nuagebec/ubuntu          Simple always updated Ubuntu docker images... 4        [OK]
docker.io  docker.io/nickistre/ubuntu-lamp-wordpress LAMP on Ubuntu with wp-cli installed         3        [OK]
docker.io  docker.io/nimmi/ubuntu             This is a docker images different LTS vers... 3        [OK]
docker.io  docker.io/densuke/ubuntu-jp-remix  Ubuntu Linuxの日本語remix風味です          2        [OK]
docker.io  docker.io/maxexcloo/ubuntu         Docker base image built on Ubuntu with Sup... 2        [OK]
docker.io  docker.io/sylvainlasnier/ubuntu    Ubuntu 15.10 root docker images with commo... 2        [OK]
docker.io  docker.io/isuper/base-ubuntu       This is just a small and clean base Ubuntu... 1        [OK]
docker.io  docker.io/nickistre/ubuntu-lamp     LAMP server on Ubuntu                       1        [OK]
docker.io  docker.io/sassmann/ubuntu-steam     Steam based on Ubuntu                       1        [OK]
docker.io  docker.io/seetheprogress/ubuntu     Ubuntu image provided by seetheprogress us... 1        [OK]
docker.io  docker.io/birkof/ubuntu            Ubuntu 14.04 LTS (Trusty Tahr)               0        [OK]
docker.io  docker.io/esyoat/ubuntu            Ubuntu LTS                                  0        [OK]
docker.io  docker.io/konstruktoid/ubuntu       Ubuntu base image                           0        [OK]
docker.io  docker.io/rallias/ubuntu            Ubuntu with the needful                      0        [OK]
docker.io  docker.io/teamrock/ubuntu           TeamRock's Ubuntu image configured with AW... 0        [OK]
docker.io  docker.io/webhippie/ubuntu          Docker images for ubuntu                     0        [OK]
docker.io  docker.io/zoni/ubuntu              Docker images for ubuntu                     0        [OK]
[root@vbox ~]#
```

Search Docker Images

7. After you decided on what image you want to run based on your needs, download it locally by running the below command (in this case an Ubuntu image is downloaded and used):

```
# docker pull ubuntu
```



```
[root@tecmint ~]# docker pull ubuntu
latest: Pulling from ubuntu

f15ce52fc004: Pull complete
c4fae638e7ce: Pull complete
a4c5be5b6e59: Pull complete
8693db7e8a00: Pull complete
ubuntu:latest: The image you are pulling has been verified. Important: image verification is a tech preview feature and should not be relied on to provide security.

Digest: sha256:457b05828bdb5dcc044d93d042863fba3f2158ae249a6db5ae3934307c757c54
Status: Downloaded newer image for ubuntu:latest
[root@tecmint ~]# _
```

Download Docker Images

8. To list all the available Docker images on your host issue the following command:

```
# docker images
```

```
[root@tecmint ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
ubuntu               latest              8693db7e8a00       7 days ago
187.9 MB
hello-world          latest              975b84d108f1       3 months ago
960 B
[root@tecmint ~]# _
```

List Docker Images

9. If you don't need a Docker image anymore and you want to remove it from the host issue the following command:

```
# docker rmi ubuntu
```

```
[root@tecmint ~]# docker rmi ubuntu
Untagged: ubuntu:latest
Deleted: 8693db7e8a0084b8aacba184cfc4ff9891924ed2270c6dec6a9d99bdcff0d1aa
Deleted: a4c5be5b6e596241b4530ade26294afa8d8a4a2b9163c30e36c87f879b0f5073
Deleted: c4fae638e7ce5e7eb92b92e6d82e3180e28b85948d362dc52a1c44a720f9b52a
Deleted: f15ce52fc004a5c3eab9128a78f7c0c2135d4f726bc54f1373120ab3ff291bcc
[root@tecmint ~]# _
```

Remove Docker Image

Step 3: Run a Docker Container

When you execute a command against an image you basically obtain a container. After the command that is executing into container ends, the container stops (you get a non-running or exited container). If you run another command into the same image again a new container is created and so on.

All the containers created will remain on the host filesystem until you choose to delete them by using the `docker rm` command.

10. In order to create and run a container, you need to run a command into a downloaded image, in this case Ubuntu, so a basic command would be to display the distribution version file inside the container using [cat command](#), as in the following example:

```
# docker run ubuntu cat /etc/issue
```

```
[root@tecmint ~]# cat /etc/issue
CentOS release 6.7 (Final)
Kernel \r on an \m

[root@tecmint ~]# docker run ubuntu cat /etc/issue
Ubuntu 14.04.3 LTS \n \l

[root@tecmint ~]# _
```

Run Docker Containers

The above command is divided as follows:

```
# docker run [local image] [command to run into container]
```

11. To run one of the containers again with the command that was executed to create it, first you must get the container ID (or the name automatically generated by Docker) by issuing the below command, which displays a list of the running and stopped (non-running) containers:

```
# docker ps -l
```

```
[root@tecmint ~]# docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
43b448b45366        ubuntu             "cat /etc/issue"    About a minute ago
Exited (0)          elegant_ptolemy
[root@tecmint ~]# _
```

List Running Docker Containers

12. Once the container ID has been obtained, you can start the container again with the command that was used to create it, by issuing the following command:

```
# docker start c629b7d70666
```

Here, the string `c629b7d70666` represents the container ID.

```
[root@tecmint ~]# docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
43b448b45366        ubuntu             "cat /etc/issue"    4 minutes ago
Exited (0) 4 minutes ago          elegant_ptolemy
[root@tecmint ~]# docker start 43b448b45366
43b448b45366
[root@tecmint ~]# _
```

Start Docker Containers

13. In case the container is running state, you can get it's ID by issuing `docker ps` command. To stop the running container issue `docker stop` command by specifying the container ID or auto-generated name.

```
# docker stop dreamy_mccarthy

# docker ps
```

```
[root@vbox ~]# docker stop dreamy_mccarthy
dreamy_mccarthy
[root@vbox ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[root@vbox ~]#
```

Start Stop Docker Containers

14. A more elegant alternative so you don't have to remember the container ID would be to allocate a unique name for every container you create by using the `--name` option on command line, as in the following example:

```
# docker run --name myname ubuntu cat /etc/debian_version
```

```
[root@vbox ~]# docker run --name myname ubuntu cat /etc/debian_version
jessie/sid
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
```

Add Name to Docker Container

15. Then, using the name that you allocated for the container, you can manipulate container (start, stop, remove, top, stats) further just by addressing its name, as in the below examples:

```
# docker start myname

# docker stats myname

# docker top myname
```

Be aware that some of the above commands might display no output if the process of command that was used to create the container finishes. When the process that runs inside the container finishes, the container stops.

Step 4: Run an Interactive Session into a Container

16. In order to interactively connect into a container shell session, and run commands as you do on any other Linux session, issue the following command:

```
# docker run -it ubuntu bash
```

```

[root@tecmin ~]# docker run -it ubuntu bash
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# ps aux
USER      PID %CPU %MEM    USZ    RSS TTY      STAT START   TIME COMMAND
root         1   0.1   0.1  18112   1972 ?        Ss   06:58   0:00 bash
root        15   0.0   0.1  15512   1144 ?        R+   06:58   0:00 ps aux
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# uname -a
Linux ff3ccd9fb856 2.6.32-573.12.1.el6.x86_64 #1 SMP Tue Dec 15 21:19:08 UTC 201
5 x86_64 x86_64 x86_64 GNU/Linux
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# w
 06:58:47 up 43 min,  0 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# cat /etc/de
debconf.conf      default/        depmod.d/
debian_version    deluser.conf
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# cat /etc/de
debconf.conf      default/        depmod.d/
debian_version    deluser.conf
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# cat /etc/debian_version
jessie/sid
;root@ff3ccd9fb856: /root@ff3ccd9fb856:/# exit
exit
[root@tecmin ~]# _

```

Start Docker Container Interactive Shell

The above command is divided as follows:

- `-i` is used to start an interactive session.
 - `-t` allocates a tty and attaches stdin and stdout.
 - `ubuntu` is the image that we used to create the container.
 - `bash` (or `/bin/bash`) is the command that we are running inside the Ubuntu container.
17. To quit and return to host from the running container session you must type `exit` command. The `exit` command terminates all the container processes and stops it.

```
# exit
```

18. If you're interactively logged on container terminal prompt and you need to keep the container in running state but exit from the interactive session, you can quit the console and return to host terminal by pressing `Ctrl+p` and `Ctrl+q` keys.


```

[root@tecmin ~]# docker run -it ubuntu bash
;root@94d7f0952f91: /root@94d7f0952f91:/# Press Ctrl+p and Ctrl+q key to go back to
;root@94d7f0952f91: /root@94d7f0952f91:/# CentOS host
;root@94d7f0952f91: /root@94d7f0952f91:/#
;root@94d7f0952f91: /root@94d7f0952f91:/# [root@tecmin ~]#
[root@tecmin ~]#
[root@tecmin ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
94d7f0952f91       ubuntu             "bash"             33 seconds ago
Up 33 seconds
[root@tecmin ~]#
[root@tecmin ~]# docker attach 94d7f0952f91 Double press Enter key to return to docker
container session
;root@94d7f0952f91: /root@94d7f0952f91:/# exit
exit
[root@tecmin ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
[root@tecmin ~]# _

```

Keep Docker Shell Session Active

19. To reconnect to the running container you need the container ID or name. Issue `docker ps` command to get the ID or name and, then, run `docker attach` command by specifying container ID or name, as illustrated in the image above:

```
# docker attach <container id>
```

20. To stop a running container from the host session issue the following command:

```
# docker kill <container id>
```

That's all for basic container manipulation. In the next tutorial we will discuss how to save, delete and run a web server into a Docker container.