



Install Docker and Learn Basic Container Manipulation in CentOS and RHEL 8/7 – Part 1

Matei Cezar | Last Updated: August 5, 2020 | Read Time: 6 mins | [CentOS](#), [Docker](#), [RedHat](#), [Virtualization](#)
| [16 Comments](#)

In this 4-article series, we will discuss Docker, which is an open-source lightweight virtualization tool that runs at top of Operating System level, allowing users to create, run and deploy applications, encapsulated into small containers.

This type of Linux containers are proven to be fast, portable, and secure. The processes that run in a Docker container are always isolated from the main host, preventing outside tampering.

Part 1: Install Docker and Learn Basic Container Manipulation in CentOS and RHEL 8/7

Part 2: [How to Deploy and Run Applications into Docker Containers on CentOS/RHEL 8/7](#)

Part 3: [Automatically Build and Configure Docker Images with Dockerfile on CentOS/RHEL 8/7](#)

Part 4: [How to Monitor Docker Containers in Linux](#)

This tutorial provides a starting point on how to install Docker, create and run Docker containers on CentOS/RHEL 8/7, but barely scratches the surface of Docker.

Step 1: Install and Configure Docker

1. Earlier versions of Docker were called `docker` or `docker-engine`, if you have these installed, you must uninstall them before installing a newer `docker-ce` version.

```
# yum remove docker \
    docker-client \
    docker-client-latest \
```

```
docker-common \
docker-latest \
docker-latest-logrotate \
docker-logrotate \
docker-engine
```

2. To install the latest version of the Docker Engine you need to set up the Docker repository and install the `yum-utils` package to enable Docker stable repository on the system.

```
# yum install -y yum-utils
# yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
```

3. Now install the newer `docker-ce` version from the Docker repository and `containerd` manually, because due to some issues, Red Hat blocked the installation of containerd.io **1.2.0-3.el7**, which is a dependency of `docker-ce`.

```
# yum install https://download.docker.com/linux/centos/7/x86_64/stable/Pack
# yum install docker-ce docker-ce-cli
```

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

```
# systemctl start docker
# systemctl status docker
# systemctl enable docker
```

```

[root@tecmint:~]# systemctl start docker
[root@tecmint:~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-08-01 06:16:13 EDT; 12min ago
     Docs: https://docs.docker.com
   Main PID: 66577 (dockerd)
    Tasks: 10
   Memory: 59.1M
   CGroup: /system.slice/docker.service
           └─66577 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 01 06:16:07 mail dockerd[66577]: time="2020-08-01T06:16:07.643266866-04:00" level=warning msg=">
Aug 01 06:16:07 mail dockerd[66577]: time="2020-08-01T06:16:07.643721515-04:00" level=info msg="Loa>
Aug 01 06:16:09 mail dockerd[66577]: time="2020-08-01T06:16:09.727244663-04:00" level=info msg="Def>
Aug 01 06:16:10 mail dockerd[66577]: time="2020-08-01T06:16:10.929837183-04:00" level=info msg="Loa>
Aug 01 06:16:12 mail dockerd[66577]: time="2020-08-01T06:16:12.062099904-04:00" level=info msg="Doc>
Aug 01 06:16:12 mail dockerd[66577]: time="2020-08-01T06:16:12.062488798-04:00" level=info msg="Dae>
Aug 01 06:16:13 mail dockerd[66577]: time="2020-08-01T06:16:13.486946647-04:00" level=info msg="API>
Aug 01 06:16:13 mail systemd[1]: Started Docker Application Container Engine.
Aug 01 06:17:28 mail dockerd[66577]: time="2020-08-01T06:17:28.418853510-04:00" level=info msg="ign>
Aug 01 06:17:41 mail dockerd[66577]: time="2020-08-01T06:17:41.034455715-04:00" level=info msg="ign>
[root@tecmint:~]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system
/docker.service.
[root@tecmint:~]# _

```

Check Docker Status

5. 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.

Sample Output

Verify Docker Installation

```

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/>

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

For system-wide information on Docker

```
# docker info
```

```
Client:
 Debug Mode: false

Server:
 Containers: 2
  Running: 0
  Paused: 0
  Stopped: 2
 Images: 1
 Server Version: 19.03.12
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 894b81a4b802e4eb2a91d1ce216b8817763c29fb
 runc version: 425e105d5a03fabd737a126ad93d62a9eeede87f
 init version: fec3683
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.18.0-193.6.3.el8_2.x86_64
 Operating System: CentOS Linux 8 (Core)
 OSType: linux
 Architecture: x86_64
 CPUs: 1
 Total Memory: 818.8MiB
:
```

[Check Docker Info](#)

For Docker version

```
# docker version
```

```
[root@tecmint:~]#  
[root@tecmint:~]# docker version  
Client: Docker Engine - Community  
Version: 19.03.12  
API version: 1.40  
Go version: go1.13.10  
Git commit: 48a66213fe  
Built: Mon Jun 22 15:46:54 2020  
OS/Arch: linux/amd64  
Experimental: false  
  
Server: Docker Engine - Community  
Engine:  
Version: 19.03.12  
API version: 1.40 (minimum version 1.12)  
Go version: go1.13.10  
Git commit: 48a66213fe  
Built: Mon Jun 22 15:45:28 2020  
OS/Arch: linux/amd64  
Experimental: false  
containerd:  
Version: 1.2.6  
GitCommit: 894b81a4b802e4eb2a91d1ce216b8817763c29fb  
runc:  
Version: 1.0.0-rc8  
GitCommit: 425e105d5a03fabd737a126ad93d62a9eeede87f  
docker-init:  
Version: 0.18.0  
GitCommit: fec3683  
[root@tecmint:~]# _
```

Check Docker Version

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

```
# docker
```

```
[root@tecmint:~]# docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/root/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides
                        --host and --tls)
  -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"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default "/root/.docker/ca.pem")
  --tlscert string      Path to TLS certificate file (default "/root/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default "/root/.docker/key.pem")
  --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
  context      Manage contexts
  image        Manage Docker images
  network      Manage Docker networks
  plugin       Manage plugins
  secret       Manage Docker secrets
  service      Manage Docker services
  stack        Manage Docker stacks
  swarm        Manage Docker swarms
  system       Manage Docker
  trust        Manage Docker trust
  volume       Manage Docker volumes
```

List Docker Commands

Step 2: Download a Docker Image

8. In order to start and run a Docker container, first, an image must be downloaded from [Docker Hub](https://hub.docker.com/) on your host. Docker Hub offers a lot of free images from its repositories.

To search for a Docker image, Ubuntu, for instance, issue the following command:

```
# docker search ubuntu
```

```
[root@tecmint:~]# docker search ubuntu
```

NAME	DESCRIPTION	STARS
ubuntu	Ubuntu is a Debian-based Linux operating sys...	11181
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	452
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offi...	246
consol/ubuntu-xfce-vnc	Ubuntu container with "headless" VNC session...	222
ubuntu-upstart	Upstart is an event-based replacement for th...	110
neurodebian	NeuroDebian provides neuroscience research s...	68
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5	ubuntu-16-nginx-php-phpmyadmin-mysql-5	50
ubuntu-debootstrap	debootstrap --variant=minbase --components=m...	44
nuagebec/ubuntu	Simple always updated Ubuntu docker images w...	24
i386/ubuntu	Ubuntu is a Debian-based Linux operating sys...	22
1and1internet/ubuntu-16-apache-php-5.6	ubuntu-16-apache-php-5.6	14
1and1internet/ubuntu-16-apache-php-7.0	ubuntu-16-apache-php-7.0	13
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...	4
1and1internet/ubuntu-16-nginx-php-7.1-wordpress-4	ubuntu-16-nginx-php-7.1-wordpress-4	3
pivotaldata/ubuntu16.04-build	Ubuntu 16.04 image for GPDB compilation	1
1and1internet/ubuntu-16-sshd	ubuntu-16-sshd	1
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development	1
1and1internet/ubuntu-16-php-7.1	ubuntu-16-php-7.1	1
smartentry/ubuntu	ubuntu with smartentry	1

```
[root@tecmint:~]#
```

Docker Search Ubuntu Images

9. 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
Using default tag: latest
latest: Pulling from library/ubuntu
3ff22d22a855: Pull complete
e7cb79d19722: Pull complete
323d0d660b6a: Pull complete
b7f616834fd0: Pull complete
Digest: sha256:5d1d5407f353843ecf8b16524bc5565aa332e9e6a1297c73a92d3e754b8a636d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[root@tecmint:~]#
```

Download Docker Ubuntu Image

10. 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             SIZE
ubuntu              latest             1e4467b07108       7 days ago         73.9MB
hello-world         latest             bf756fb1ae65       7 months ago       13.3kB
[root@tecmint:~]#
```

List Docker Images

11. 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
Untagged: ubuntu@sha256:5d1d5407f353843ecf8b16524bc5565aa332e9e6a1297c73a92d3e754b8a636d
Deleted: sha256:1e4467b07108685c38297025797890f0492c4ec509212e2e4b4822d367fe6bc8
Deleted: sha256:7515ee845913c8df9826c988341a09e0240e291c66bdc436a067e070d7910a1f
Deleted: sha256:50ebe6a0675f1ed7ca499a2ec7d8cc993d495dd66ca1035c218ec5efcb6fbb8c
Deleted: sha256:2515e0ecfb82d58c004c4b53fcf9230d9eca8d0f5f823c20172be01eec587ccb
Deleted: sha256:ce30112909569cead47eac188789d0cf95924b166405aa4b71fb500d6e4ae08d
[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 the 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.

12. In order to create and run a container, you need to run 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:~]# docker run ubuntu cat /etc/issue
Ubuntu 20.04 LTS \n \l

[root@tecmint:~]# _
```

Run Docker Containers

The above command is divided as follows:

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

13. 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
923a720da57f   ubuntu   "cat /etc/issue"        2 minutes ago   Exited (0) 2 minutes ago
[root@tecmint:~]# _
```

List Running Docker Containers

14. 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 923a720da57f
```

Here, the string **923a720da57f** represents the container ID.

```
[root@tecmint:~]# docker ps -l
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
923a720da57f   ubuntu   "cat /etc/issue"        2 minutes ago   Exited (0) 2 minutes ago
[root@tecmint:~]#
[root@tecmint:~]# docker start 923a720da57f
923a720da57f
[root@tecmint:~]# _
```

Start Docker Containers

15. In case the container is running state, you can get its 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 923a720da57f
OR
# docker stop cool_lalande
# docker ps
```

```
[root@tecmint:~]# docker ps -l
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
923a720da57f   ubuntu    "cat /etc/issue"        2 minutes ago Exited (0) 2 minutes ago          cool_lalande
[root@tecmint:~]#
[root@tecmint:~]# docker start 923a720da57f
923a720da57f
[root@tecmint:~]# docker stop 923a720da57f
923a720da57f
[root@tecmint:~]# docker stop cool_lalande
cool_lalande
[root@tecmint:~]# _
```

Stop Docker Containers

16. 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 the command line, as in the following example:

```
# docker run --name ubuntu20.04 ubuntu cat /etc/issue
```

```
[root@tecmint:~]# docker run --name ubuntu20.04 ubuntu cat /etc/issue
Ubuntu 20.04 LTS \n \l
[root@tecmint:~]# _
```

Add Name to Docker Container

17. 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 ubuntu20.04
# docker stats ubuntu20.04
# docker top ubuntu20.04
```

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

18. 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@tecmint:~]# docker run -it ubuntu bash
root@987b92f99b27:/# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  1.2  0.4   4112   3476 pts/0    Ss   11:04   0:00 bash
root           8  0.0  0.3   5884   2904 pts/0    R+   11:05   0:00 ps aux
root@987b92f99b27:/#
root@987b92f99b27:/# uname -a
Linux 987b92f99b27 4.18.0-193.6.3.el8_2.x86_64 #1 SMP Wed Jun 10 11:09:32 UTC 2020 x86_64
root@987b92f99b27:/#
root@987b92f99b27:/# w
 11:05:18 up  5:32,  0 users,  load average: 0.07, 0.08, 0.09
USER          TTY          FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
root@987b92f99b27:/#
root@987b92f99b27:/# cat /etc/issue
Ubuntu 20.04 LTS \n \l
root@987b92f99b27:/# _
```

Start Docker Container Interactive Shell

The above command is divided as follows:

1. `-i` is used to start an interactive session.
2. `-t` allocates a TTY and attaches stdin and stdout.
3. `ubuntu` is the image that we used to create the container.
4. `bash` (or `/bin/bash`) is the command that we are running inside the Ubuntu container.

19. 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
```

20. 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@tecmint ~]# 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@tecmint ~]#
[root@tecmint ~]#
[root@tecmint ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
94d7f0952f91       ubuntu             "bash"             33 seconds ago
Up 33 seconds
[root@tecmint ~]#
[root@tecmint ~]# docker attach 94d7f0952f91 Double press Enter key to return to docker
container session
[root@94d7f0952f91: /root@94d7f0952f91:/# exit
exit
[root@tecmint ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
[root@tecmint ~]# _
```

Keep Docker Shell Session Active

21. 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>
```

22. 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.

► [CentOS Tips](#), [Docker Tips](#), [RHEL Tips](#), [virtualization](#)

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth \$20!

Learn [more about the contest](#) and stand a chance to win by [sharing your thoughts below](#)!

GIVEAWAY!

Win eBooks



www.tecmint.com

PREVIOUS ARTICLE:

[Marcel – A More Modern Shell for Linux](#)

NEXT ARTICLE:

How to Install, Run and Delete Applications Inside Docker Containers – Part 2



Matei Cezar

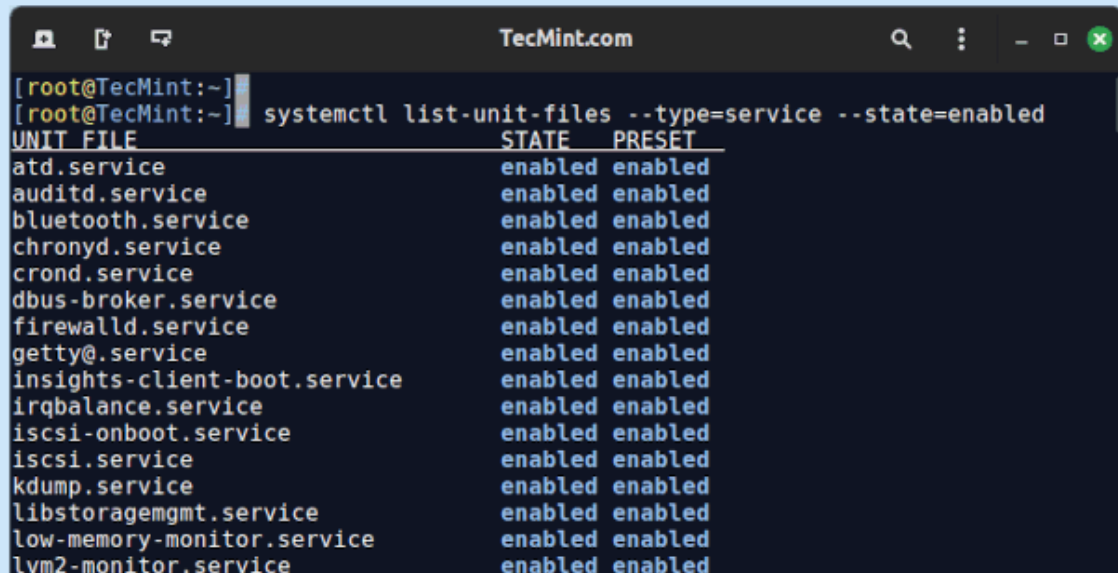
I'm a computer addicted guy, a fan of open source and linux based system software, have about 4 years experience with Linux distributions desktop, servers and bash scripting.

Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality standards.

Join the [TecMint Weekly Newsletter](#) (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please [add a comment](#) or [buy me a coffee](#) to show your appreciation.

Related Posts



```
[root@TecMint:~]# systemctl list-unit-files --type=service --state=enabled
```

UNIT FILE	STATE	PRESET
atd.service	enabled	enabled
auditd.service	enabled	enabled
bluetooth.service	enabled	enabled
chronyd.service	enabled	enabled
crond.service	enabled	enabled
dbus-broker.service	enabled	enabled
firewalld.service	enabled	enabled
getty@.service	enabled	enabled
insights-client-boot.service	enabled	enabled
irqbalance.service	enabled	enabled
iscsi-onboot.service	enabled	enabled
iscsi.service	enabled	enabled
kdump.service	enabled	enabled
libstoragemgmt.service	enabled	enabled
low-memory-monitor.service	enabled	enabled
lvm2-monitor.service	enabled	enabled

How to Disable and Remove Unnecessary Services on Linux



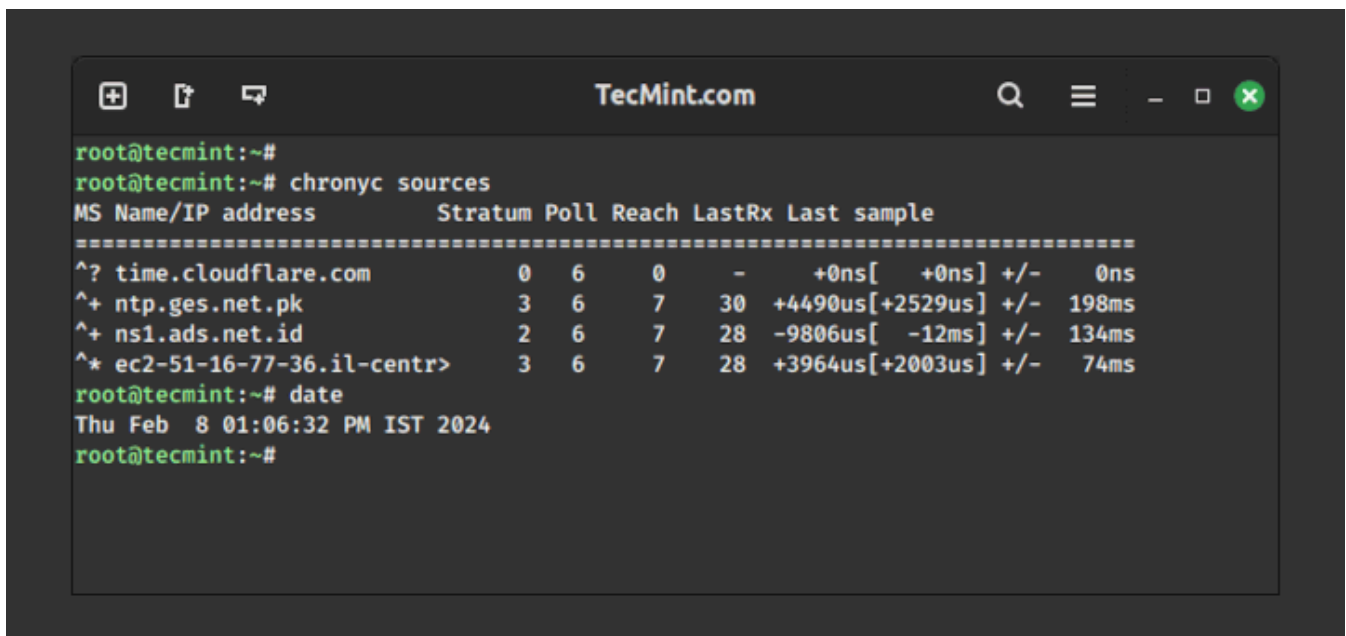
Improve Website Performance – Install Memcached on RHEL 9



How to Migrate CentOS 7 to Rocky Linux 9



How to Reset Forgotten Root Password in RHEL Systems



```
root@tecmint:~#  
root@tecmint:~# chronyc sources  
MS Name/IP address          Stratum Poll Reach LastRx Last sample  
=====
```

MS Name/IP address	Stratum	Poll	Reach	LastRx	Last sample
^? time.cloudflare.com	0	6	0	-	+0ns[+0ns] +/- 0ns
^+ ntp.ges.net.pk	3	6	7	30	+4490us[+2529us] +/- 198ms
^+ ns1.ads.net.id	2	6	7	28	-9806us[-12ms] +/- 134ms
^* ec2-51-16-77-36.il-centr>	3	6	7	28	+3964us[+2003us] +/- 74ms

```
root@tecmint:~# date  
Thu Feb  8 01:06:32 PM IST 2024  
root@tecmint:~#
```

How to Install and Configure NTP in RHEL Systems



How to Install Terraform (Infrastructure as Code) in Linux

 **16 Comments**

[Leave a Reply](#)

jagadeesh

July 11, 2018 at 12:00 am

```
docker run hello-world
```

Unable to find image 'hello-world:latest' locally

Pulling repository hello-world

Get <https://index.docker.io/v1/repositories/library/hello-world/images>: x509: certificate has expired or is not yet valid

Am getting this error what to do sir? i like to study these tech by manually

[Reply](#)

Admin



Ravi Saive

July 11, 2018 at 10:53 am

@Jagadeesh,

I think you behind a firewall, and you need to configure proxy to make it work.

Please add the proxy settings in `/etc/default/docker` file, and restarted the service. Then try to run the hello-world image and do further tasks.

[Reply](#)

skv20224

May 23, 2018 at 5:37 pm

Sir you are really awesome, whenever I get any kind of problem or troubleshooting i always first search your articles at [Tecmint.com](https://www.tecmint.com).

[Reply](#)

Gaurav Bhandari

March 1, 2018 at 10:31 pm

Thanks for info :) I always have to use sudo docker but I can live with it :)

[Reply](#)

saik

May 23, 2017 at 8:52 pm

```
[root@localhost ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 20G 0 disk
├─sda1 8:1 0 300M 0 part /boot
├─sda2 8:2 0 2G 0 part [SWAP]
└─sda3 8:3 0 17.7G 0 part /
sr0 11:0 1 1024M 0 rom
loop0 7:0 0 100G 0 loop
└─docker-8:3-34878977-pool 253:0 0 100G 0 dm
loop1 7:1 0 2G 0 loop
└─docker-8:3-34878977-pool 253:0 0 100G 0 dm
[root@localhost ~]#
```

[Reply](#)**saik**

May 23, 2017 at 8:51 pm

Hello Sir,

I installed docker container default it was created with a size of 100G now how to remove by using cmd.

See the attachment I'm giving below sir...

[Reply](#)**Chinmay**

February 24, 2017 at 11:47 am

Hi,

I need to run multiple web sites on a single server. Each website will have different version of php. Please let me know whether a docker container will be available for such requirement or I have to create such containers on my own.

If I need to create containers with different version of php, please guide me for the same as I am starting on docker today only. thanks in advance.

[Reply](#)**Matei Cezar**

February 24, 2017 at 12:19 pm

This can be easily be accomplished with Vhosts. If you really want containers then you need to create a container for each application or version of php and expose it to outside.

[Reply](#)

david

March 3, 2016 at 10:55 pm

Hi – A genuine question please. I use Proxmox for LXC containers which works great..what does Docker offer over proxmox ?

[Reply](#)



Matei Cezar

March 4, 2016 at 2:07 pm

Basically, I would say that the main difference is that LXC containers behave as a virtual machine mainly, opposed to Docker which runs an ephemeral single process. But that's only just a simple difference. You should google it for more detailed info.

[Reply](#)

alex

February 13, 2016 at 11:04 pm

Thanks for the article..

I completed a 6 month contract for a large publisher involving the evaluation of various triple stores. I created a dockerfile for each triple store allowing us to quickly install the systems when required on various servers. I also created a system to monitor and graph the memory use of any number docker containers. They are a number of stumbling blocks that takes some experience to understand.

Regards

Alex

<http://Www.tilogeo.com>

[Reply](#)

Raghu

October 24, 2016 at 11:21 pm

Seems tilogeo.com link is down & unavailable.

[Reply](#)

Bangaru Adabala

January 15, 2018 at 4:54 pm

Alex, If you have the details on how you accomplished these docker tasks, can you please share on my gmail e-mail ID bangaru.adabala I was trying to create some docker images with personal scripts and other stuff to be able to quickly deploy and / or restore instances.

Thanks in advance.

[Reply](#)**Shambhu Rajput**

February 7, 2016 at 12:39 pm

Thank you

[Reply](#)**Matei Cezar**

February 3, 2016 at 8:46 pm

You could use Docker on any machine, it doesn't matter if you are running LAMP stack on it. As a recommendation try to run Docker on a fresh server or a test server, never play on production machines for the sake of security and resources.

[Reply](#)**Shambhu**

February 2, 2016 at 3:24 pm

Hi,

For docker installation we need fresh OS or can install at running LAMP server?

Regard

[Reply](#)

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our [comment policy](#) to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

[Buy Me a Coffee](#)

Linux Commands and Tools

[How to Use 'head' Command in Linux \[8 Useful Examples\]](#)

[How to Save Command Output to a File in Linux](#)

[How to Create and Extract Zip Files to Specific Directory in Linux](#)

[How to Use 'tee' Command in Linux \[8 Useful Examples\]](#)

[Manage Log Messages Under Systemd Using Journalctl \[Comprehensive Guide\]](#)

[How to Install Particular Versions of Packages with Snap](#)

Linux Server Monitoring Tools

[BpyTop – Resource Monitoring Tool for Linux](#)

[Hardinfo2 – Check Hardware Information in Linux](#)

[Icinga: A Next Generation Open Source ‘Linux Server Monitoring’ Tool for RHEL/CentOS 7.0](#)

[ngrep – A Network Packet Analyzer for Linux](#)

[Nethogs – Monitor Linux Network Traffic Usage Per Process](#)

[How to Install Cacti with Cacti-Spine in Debian and Ubuntu](#)

Learn Linux Tricks & Tips

[Powerline – Adds Statuslines and Prompts to Vim and Bash Shell](#)

[4 Ways to Find Server Public IP Address in Linux Terminal](#)

[6 Best CLI Tools to Search Plain-Text Data Using Regular Expressions](#)

[How to Copy File Permissions and Ownership to Another File in Linux](#)

[4 Useful Tips on mkdir, tar and kill Commands in Linux](#)

[How to Find and Sort Files Based on Modification Date and Time in Linux](#)

Best Linux Tools

[5 Best Command Line HTTP Clients for Linux](#)

[6 Best Linux Boot Loaders](#)

[8 Best Command-Line/Terminal Email Clients for Linux](#)

[16 Best Microsoft Teams Alternatives For Linux in 2024](#)

[8 Best Open-Source Disk Cloning & Backup Tools for Linux \(2024\)](#)

[10 Best Tools to Install on Fresh Linux Mint Installation](#)

Tecmint: Linux Howtos, Tutorials & Guides © 2024. All Rights Reserved.

The material in this site cannot be republished either online or offline, without our permission.

Hosting Sponsored by : [Linode Cloud Hosting](#)