

yum update

sudo tee /etc/yum.repos.d/docker.repo <<-'EOF'

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main/centos/7/

enabled=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg

EOF

sudo yum install docker-engine

sudo service docker start

docker run hello-world

**Install with the script**

1. Log into your machine as a user with sudo or root privileges.
2. Make sure your existing yum packages are up-to-date.
3. $ sudo yum update
4. Run the Docker installation script.
5. $ curl -fsSL https://get.docker.com/ | sh

This script adds the docker.repo repository and installs Docker.

1. Start the Docker daemon.
2. $ sudo service docker start
3. Verify docker is installed correctly by running a test image in a container.
4. $ sudo docker run hello-world
5. Log into Centos as a user with sudo privileges.
6. Create the docker group.

sudo groupadd docker

1. Add your user to docker group.

sudo usermod -aG docker your\_username

1. Log out and log back in.

This ensures your user is running with the correct permissions.

1. Verify your work by running docker without sudo.
2. $ docker run hello-world

chkconfig docker on

# Centos

To get the latest stable official CentOS image on [Docker Hub](https://registry.hub.docker.com/_/centos/):

$ sudo docker pull centos

This command only pulls the image tagged centos:latest, which always points to the latest stable CentOS release, currently CentOS 7 (*centos:centos7*). To pull any other version of the CentOS image, for example CentOS 6:

$ sudo docker pull centos:centos6

docker images centos

docker run centos:latest cat /etc/centos-release

docker ps

docker run –it <container name> /bin/bash

docker run <image name> bash

 **$ docker images**

 Pulling an *alpine* image:  
  
**$ docker pull alpine**

 Run a container from a locally-available image:  
  
**$ docker run -it alpine sh**

 Run a container in the background (*-d option*) from an image :  
  
**$ docker run -d nginx**

 List only running containers:  
  
**$ docker ps**

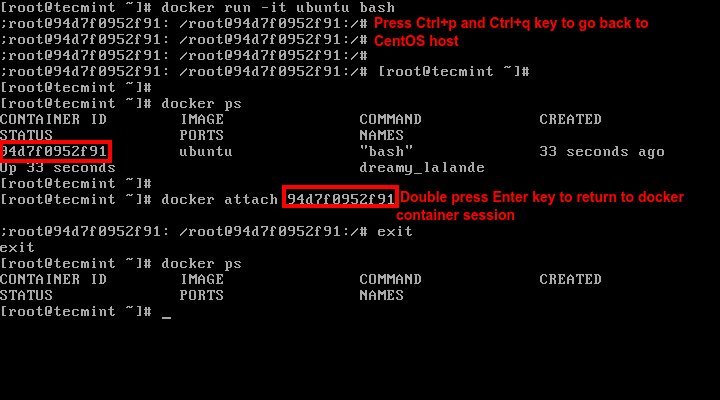
 List all containers:   
  
**$ docker ps -a**

 Inject a process inside a running container:  
  
**$ docker exec -it <container\_id/name> bash**

 Stop a container:  
  
**$ docker stop <container id/name>**

 Delete a container:  
  
**$ docker rm  <container id/name>**

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.

[](http://www.tecmint.com/wp-content/uploads/2016/01/Keep-Docker-Shell-Session.png)

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 manipu

# Uninstall

You can uninstall the Docker software with yum.

1. List the package you have installed.
2. $ yum list installed | grep docker
3. yum list installed | grep docker
4. docker-engine.x86\_64 1.7.1-1.el7 @/docker-engine-1.7.1-1.el7.x86\_64.rpm
5. Remove the package.
6. $ sudo yum -y remove docker-engine.x86\_64

This command does not remove images, containers, volumes, or user-created configuration files on your host.

1. To delete all images, containers, and volumes, run the following command:
2. $ rm -rf /var/lib/docker

Container orchestration is an umbrella term which encompasses container scheduling and cluster management.