

Docker Installation

apt update	Updates the package sources with the latest versions
apt install docker.io -y	Install docker in Ubuntu Linux distribution
systemctl start docker	Start docker services
systemctl stop docker	Stop docker services
systemctl restart docker	Stop and start docker services
systemctl status docker	Check the status of docker services

Container Operations

docker container create --name web -it alpine sh	Create a named container
docker container start web	Start a container
docker container stop web	Stop a container
docker container restart web	stop and start a container
docker container run -it --name myweb alpine sh	create and start a named container
Ctrl + p, Ctrl + q	Detach from the container
docker container attach myweb	Attach to a running container
docker container logs myweb	See the logs of the container
docker container rename myweb myalpine	Change the name of a container
docker container run -h alpine -it --rm alpine sh	Set the hostname of a container
docker container run -it -w /var --rm alpine sh	Set the current working directory
docker container run -it --env "WEB_HOST=172.168.1.1" --rm alpine sh	Set the environment variables

Container Operations (cont)

docker container exec myalpine [command]	Create any process inside an already running container
--	--

Docker Management

docker version	prints the current version number
docker info	displays system wide information
docker login	Log in to a DockerHub registry
docker logout	Logout from DockerHub registry
docker inspect [object]	Display detailed information on one or more objects.
docker --help	list the help on any command
docker image [option]	To work with images
docker container [option]	To work with containers
docker volume [option]	To work with volumes
docker network [option]	To work with network

List

docker image ls	List of images
docker image ls --digests	List of image digests
docker container ls	List of running containers
docker container ls -a	List of all containers
docker container ls -q	Get only the containers IDs of containers
docker volume ls	List of volumes
docker network ls	List of networks



By pratikshahp

Published 10th March, 2023.

Last updated 20th March, 2023.

Page 1 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Docker Images

<code>docker search nginx</code>	List the available images on DockerHub
<code>docker image pull nginx</code>	Pull an image from DockerHub
<code>docker image pull myregistry.com:8080/alpine:latest</code>	Pull an image from a private registry
<code>vim Dockerfile</code> <code>FROM alpine</code> <code>RUN date > /data</code>	Dockerfile to build an image
<code>docker image build -t php/alpine:dockerfile .</code>	Create an image from a Dockerfile
<code>docker container run -it --name myimage alpine sh</code> <code>/ # date > /data</code> <code>/ # cat /data</code>	Run a container and modify it.
<code>docker container diff myimage</code>	Check the difference with the base image
<code>docker container commit myimage php/alpine:test</code>	Save a running container as an image.
<code>docker container run -it php/alpine:test sh</code> <code>cat /data</code>	Run the created image as a container
<code>docker image push php/alpine:test</code>	Push the image to DockerHub registry

Docker Volumes

<code>docker container run -itv /data --name myvol alpine sh</code>	Create a directory on the host system and mount it under /data of the container
<code>docker volume create --name myvolume</code>	Create a named volume
<code>docker container run -itv myvolume:/data --name cmyvol alpine sh</code>	Mount a named volume
<code>docker system df</code>	Get the information about disk storage

Docker Networking

<code>docker container run -d --name myweb -p 8080:80 nginx</code>	Access the container from the external world
<code>docker container run -d --name myweb1 -P nginx:alpine</code>	Expose a container port to a random port of the host system
<code>docker container run -it --network=none alpine sh</code>	To Request Docker daemon not to create an interface for the container
<code>docker container run -it --network=host alpine sh</code>	share the network namespace of the host system with the container
<code>docker container run -it --network=container:myweb1 alpine sh</code>	Share the network namespace of an existing container
<code>docker network create mynetwork</code>	Create a user defined network
<code>docker container run -d --name nginxnet --network=mynetwork nginx:alpine</code>	Create a container and attach it to the network
<code>docker network connect mynetwork web</code>	Attach a running container to a network
<code>docker network disconnect mynetwork web</code>	Disconnect from a specific network

Docker Resource Allocation

<code>docker container run -it --ulimit nproc=10 --rm alpine sh</code>	Set the maximum number of processes the container can create
<code>docker container run -d --name cpulimit --cpuset-cpus="0" alpine top</code>	Container can run only on CPU "0".
<code>docker container run -d --name memlimit --memory "200m" alpine top</code>	Limit the maximum amount of memory usage



By **pratikshahp**

Published 10th March, 2023.

Last updated 20th March, 2023.

Page 2 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Docker Compose

<code>curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose</code>	Install a docker compose
<code>chmod +x /usr/local/bin/docker-compose</code>	Set the executable
<code>mkdir wordpress</code> <code>cd wordpress/</code> <code>vim docker-compose.yml</code>	Create a folder and yaml file
<code>docker-compose up</code>	Start the application in the foreground
<code>docker-compose up -d</code>	Start the application in the background
<code>docker-compose ps</code>	List containers created by the application
<code>docker-compose stop</code>	Stop all the containers for the given application

Clean Up (cont)

<code>docker-compose rm</code>	Remove the containers for the given application
<code>docker-compose down -v</code>	Delete the application completely with its containers, networks, volumes and images
<code>docker system prune</code>	Remove unused images, stopped containers, and unused networks

Clean Up

<code>docker image rm nginx:alpine</code>	Remove an image locally
<code>docker image prune</code>	Clean an unused/dangling image
<code>docker container rm -f web</code>	Delete a running container
<code>docker container rm myweb</code>	Delete a stopped container
<code>docker container prune</code>	Remove all stopped containers
<code>docker network rm mynetwork</code>	Remove a specific network
<code>docker network prune</code>	Remove all networks which are not referenced by any containers
<code>docker volume rm myvolume</code>	Remove a specific volume
<code>docker volume prune</code>	Remove all unused volume



By pratikshahp

Published 10th March, 2023.

Last updated 20th March, 2023.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>