

Lab: Installing Docker and Managing Docker Containers

```
root@myVM: ~  
login as: anusha954  
anusha954@20.70.141.13's password:  
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Tue Aug 13 04:10:24 UTC 2024  
  
System load:  0.08      Processes:      117  
Usage of /:   5.0% of 28.02GB  Users logged in:  0  
Memory usage: 3%      IPv4 address for eth0: 10.0.0.4  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
anusha954@myVM:~$ sudo su -  
root@myVM:~# dnf -y install yum-utils  
root@myVM:~# sudo apt-get update  
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]  
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:7 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]  
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]  
Get:9 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]  
Get:10 http://azure.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]  
Get:11 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]  
Get:12 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
```

—  

```
root@myVM: ~
Get:41 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [245 k
B]
Get:42 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted Translation-en [47.8
kB]
Get:43 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6
kB]
Get:44 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808
B]
Get:45 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208
B]
Get:46 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata
[344 B]
Fetched 28.3 MB in 6s (5108 kB/s)
Reading package lists... Done
root@myVM:~# sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.2).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@myVM:~# sudo install -m 0755 -d /etc/apt/keyrings
root@myVM:~# sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyring
s/docker.asc
root@myVM:~# sudo chmod a+r /etc/apt/keyrings/docker.asc
root@myVM:~# echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://dow
nload.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
root@myVM:~# sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [11.8 kB]
Fetched 60.6 kB in 1s (84.6 kB/s)
Reading package lists... Done
root@myVM:~# sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras libbtdl7 libslirp0 pigz slirp4netns
Suggested packages:
```

```
root@myVM: ~  
The following additional packages will be installed:  
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns  
Suggested packages:  
  aufs-tools cgroupfs-mount | cgroup-lite  
The following NEW packages will be installed:  
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras  
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns  
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.  
Need to get 122 MB of archives.  
After this operation, 437 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]  
Get:2 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [4  
0.3 kB]  
Get:3 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3  
[63.8 kB]  
Get:4 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1bu  
ild2 [34.9 kB]  
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.20-  
1 [30.5 MB]  
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64  
0.16.1-1-ubuntu.24.04~noble [29.9 MB]  
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.1.  
1-1-ubuntu.24.04~noble [14.6 MB]  
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.1.1-1-  
ubuntu.24.04~noble [25.3 MB]  
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras a  
md64 5:27.1.1-1-ubuntu.24.04~noble [9318 kB]  
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd6  
4 2.29.1-1-ubuntu.24.04~noble [12.5 MB]  
Fetched 122 MB in 2s (75.5 MB/s)  
Selecting previously unselected package pigz.  
(Reading database ... 64525 files and directories currently installed.)  
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...  
Unpacking pigz (2.8-1) ...  
Selecting previously unselected package containerd.io.  
Preparing to unpack .../1-containerd.io_1.7.20-1_amd64.deb ...  
Unpacking containerd.io (1.7.20-1) ...  
Selecting previously unselected package docker-buildx-plugin.  
Preparing to unpack .../2-docker-buildx-plugin_0.16.1-1-ubuntu.24.04~noble_amd64.deb ...  
Unpacking docker-buildx-plugin (0.16.1-1-ubuntu.24.04~noble) ...  
Selecting previously unselected package docker-ce-cli.  
Preparing to unpack .../3-docker-ce-cli_5%3a27.1.1-1-ubuntu.24.04~noble_amd64.deb ...  
Unpacking docker-ce-cli (5:27.1.1-1-ubuntu.24.04~noble) ...  
Selecting previously unselected package docker-ce.  
Preparing to unpack .../4-docker-ce_5%3a27.1.1-1-ubuntu.24.04~noble_amd64.deb ...  
Unpacking docker-ce (5:27.1.1-1-ubuntu.24.04~noble) ...  
Selecting previously unselected package docker-ce-rootless-extras.  
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.1.1-1-ubuntu.24.04~noble_amd64.deb
```

```
root@myVM: ~  
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.1.1-1~ubuntu.24.04~noble_amd64.deb  
...  
Unpacking docker-ce-rootless-extras (5:27.1.1-1~ubuntu.24.04~noble) ...  
Selecting previously unselected package docker-compose-plugin.  
Preparing to unpack .../6-docker-compose-plugin_2.29.1-1~ubuntu.24.04~noble_amd64.deb ...  
Unpacking docker-compose-plugin (2.29.1-1~ubuntu.24.04~noble) ...  
Selecting previously unselected package libltdl7:amd64.  
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...  
Unpacking libltdl7:amd64 (2.4.7-7build1) ...  
Selecting previously unselected package libslirp0:amd64.  
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...  
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...  
Selecting previously unselected package slirp4netns.  
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...  
Unpacking slirp4netns (1.2.1-1build2) ...  
Setting up docker-buildx-plugin (0.16.1-1~ubuntu.24.04~noble) ...  
Setting up containerd.io (1.7.20-1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/sys  
temd/system/containerd.service.  
Setting up docker-compose-plugin (2.29.1-1~ubuntu.24.04~noble) ...  
Setting up libltdl7:amd64 (2.4.7-7build1) ...  
Setting up docker-ce-cli (5:27.1.1-1~ubuntu.24.04~noble) ...  
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...  
Setting up pigz (2.8-1) ...  
Setting up docker-ce-rootless-extras (5:27.1.1-1~ubuntu.24.04~noble) ...  
Setting up slirp4netns (1.2.1-1build2) ...  
Setting up docker-ce (5:27.1.1-1~ubuntu.24.04~noble) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd  
/system/docker.service.  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/sys  
tem/docker.socket.  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@myVM:~# sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
c1ec31eb5944: Pull complete
```



 root@myVM: ~

```
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
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/>

```
root@myVM:~# docker
```

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

A self-sufficient runtime for containers

Common Commands:

run	Create and run a new container from an image
exec	Execute a command in a running container
ps	List containers
build	Build an image from a Dockerfile
pull	Download an image from a registry
push	Upload an image to a registry
images	List images
login	Log in to a registry
logout	Log out from a registry
search	Search Docker Hub for images
version	Show the Docker version information
info	Display system-wide information

Management Commands:

builder	Manage builds
buildx*	Docker Buildx

29°C Partly sunny Search 10:34 AM 13-08-2024

 root@myVM: ~

```
builder      Manage builds
buildx*      Docker Buildx
compose*     Docker Compose
container    Manage containers
context      Manage contexts
image        Manage images
manifest     Manage Docker image manifests and manifest lists
network      Manage networks
plugin       Manage plugins
system       Manage Docker
trust        Manage trust on Docker images
volume       Manage volumes
```

Swarm Commands:

swarm Manage Swarm

Commands:

attach	Attach local standard input, output, and error streams to a running container
commit	Create a new image from a container's changes
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
diff	Inspect changes to files or directories on a container's filesystem
events	Get real time events from the server
export	Export a container's filesystem as a tar archive
history	Show the history of an image
import	Import the contents from a tarball to create a filesystem image
inspect	Return low-level information on Docker objects
kill	Kill one or more running containers
load	Load an image from a tar archive or STDIN
logs	Fetch the logs of a container
pause	Pause all processes within one or more containers
port	List port mappings or a specific mapping for the container
rename	Rename a container
restart	Restart one or more containers
rm	Remove one or more containers
rmi	Remove one or more images
save	Save one or more images to a tar archive (streamed to STDOUT by default)
start	Start one or more stopped containers
stats	Display a live stream of container(s) resource usage statistics
stop	Stop one or more running containers
tag	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top	Display the running processes of a container
unpause	Unpause all processes within one or more containers
update	Update configuration of one or more containers
wait	Block until one or more containers stop, then print their exit codes

Global 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
```

29°C Partly sunny Search 10:34 AM 13-08-2024

```
root@myVM: ~  
  
        (overrides DOCKER_HOST env var and default context set with  
        "docker context use")  
-D, --debug          Enable debug mode  
-H, --host list      Daemon socket 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")  
    --tlsctert 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  
  
Run 'docker COMMAND --help' for more information on a command.  
  
For more help on how to use Docker, head to https://docs.docker.com/go/guides/  
root@myVM:~# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)  
   Active: active (running) since Tue 2024-08-13 04:43:45 UTC; 6min ago  
TriggeredBy: ● docker.socket  
   Docs: https://docs.docker.com  
  Main PID: 2874 (dockerd)  
    Tasks: 11  
   Memory: 28.8M (peak: 30.8M)  
    CPU: 645ms  
   CGroup: /system.slice/docker.service  
           └─2874 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
Aug 13 04:43:44 myVM dockerd[2874]: time="2024-08-13T04:43:44.653586621Z" level=info msg="St  
Aug 13 04:43:44 myVM dockerd[2874]: time="2024-08-13T04:43:44.656669452Z" level=info msg="de  
Aug 13 04:43:44 myVM dockerd[2874]: time="2024-08-13T04:43:44.779973605Z" level=info msg="Lo  
Aug 13 04:43:45 myVM dockerd[2874]: time="2024-08-13T04:43:45.263426455Z" level=info msg="Lo  
Aug 13 04:43:45 myVM dockerd[2874]: time="2024-08-13T04:43:45.290701828Z" level=warning msg=  
Aug 13 04:43:45 myVM dockerd[2874]: time="2024-08-13T04:43:45.290966830Z" level=info msg="Do  
Aug 13 04:43:45 myVM dockerd[2874]: time="2024-08-13T04:43:45.291136632Z" level=info msg="De  
Aug 13 04:43:45 myVM dockerd[2874]: time="2024-08-13T04:43:45.370857428Z" level=info msg="AF  
Aug 13 04:43:45 myVM systemd[1]: Started docker.service - Docker Application Container Engin  
Aug 13 04:44:13 myVM dockerd[2874]: time="2024-08-13T04:44:13.409416919Z" level=info msg="ig  
    ESCOC  
r Engine  
ocker.service; enabled; preset: enabled)  
8-13 04:43:45 UTC; 6min ago
```



 root@myVM: ~

```
--containerd=/run/containerd/containerd.sock
-08-13T04:43:44.653586621Z level=info msg="Starting up"
-08-13T04:43:44.656669452Z level=info msg="detected 127.0.0.53 nameserver, assuming systemd"
-08-13T04:43:44.779973605Z level=info msg="Loading containers: start."
-08-13T04:43:45.263426455Z level=info msg="Loading containers: done."
-08-13T04:43:45.290701828Z level=warning msg="Not using native diff for overlay2, this may be slower"
-08-13T04:43:45.290966830Z level=info msg="Docker daemon" commit=ccl3f95 containerd-snapshotter=v1.5.0-rc2
-08-13T04:43:45.291136632Z level=info msg="Daemon has completed initialization"
-08-13T04:43:45.370857428Z level=info msg="API listen on /run/docker.sock"
r.service - Docker Application Container Engine.
-08-13T04:44:13.409416919Z level=info msg="ignoring event" container=9a11591fb226f2646374d6
```

```
root@myVM:~# docker container create nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
efc2b5ad9ec: Pull complete
8fe9a55eb80f: Pull complete
045037a63be8: Pull complete
7111b42b4bfa: Pull complete
3dfc528a4df9: Pull complete
9e891cdb453b: Pull complete
0f11e17345c5: Pull complete
Digest: sha256:6af79ae5de40728dcea8b00d5c37ace95441fd58a8bd2aaled93f5511bb18c
```

29°C Partly sunny Search 10:35 AM 13-08-2024

 root@myVM: ~

```
-08-13T04:43:45.370857428Z" level=info msg="API listen on /run/docker.sock"
r.service - Docker Application Container Engine.
-08-13T04:44:13.409416919Z" level=info msg="ignoring event" container=9a11591fb226fb2646374f
```

```
root@myVM:~# docker container create nginx
```

```
Unable to find image 'nginx:latest' locally
```

```
latest: Pulling from library/nginx
```

```
efc2b5ad9eec: Pull complete
```

```
8fe9a55eb80f: Pull complete
```

```
045037a63be8: Pull complete
```

```
7111b42b4bfa: Pull complete
```

```
3dfc528a4df9: Pull complete
```

```
9e891cdb453b: Pull complete
```

```
0f11e17345c5: Pull complete
```

Digest: sha256:6af79ae5de407283dcea8b00d5c37ace95441fd58a8b1d2aa1ed93f5511bb18c


```
Status: Downloaded newer image for nginx:latest
```

b6ad522700512ed37215d75be6b8205550a8ef19bc29ef9ed7ee7a6c4b5941d9

```
root@myVM:~# docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
b6ad52270051	nginx	"/docker-entrypoint..."	19 seconds ago	Created
9a11591fb226	trusting_bouman	hello-world	7 minutes ago	Exited (0) 7 minutes ago
go	charming_solomon	"/hello"		

```
root@myVM:~#
```

 1 29°C

Partly sunny

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
efc2b5ad9eec: Pull complete
8fe9a55eb90f: Pull complete
045037a63be8: Pull complete
7111b42b4bfa: Pull complete
3dfc528a4df9: Pull complete
9e891cdb453b: Pull complete
0f11e17345c5: Pull complete
Digest: sha256:6af79ae5de407283dcea8b00d5c37ace95441fd58a8b1d2aa1ed93f5511bb18c
Status: Downloaded newer image for nginx:latest
b6ad522700512ed37215d75be6b8205550a8ef19bc29ef9ed7ee7a6c4b5941d9
root@myVM:~# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
b6ad52270051   nginx         "/docker-entrypoint..." 19 seconds ago Created
trusting bouman
9a11591fb226   hello-world   "/hello"                  7 minutes ago  Exited (0) 7 minutes a
go            charming solomon
root@myVM:~# # docker container create --name web01 nginx
root@myVM:~# docker container create --name web01 nginx
5e360c47e659149bbd74308cb30311573a61c12acc53cef0f4e36bf25d4db480
root@myVM:~# docker containe ls -a
unknown shorthand flag: 'a' in -a
See 'docker --help'.
```

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:

run	Create and run a new container from an image
exec	Execute a command in a running container
ps	List containers
build	Build an image from a Dockerfile
pull	Download an image from a registry
push	Upload an image to a registry
images	List images
login	Log in to a registry
logout	Log out from a registry
search	Search Docker Hub for images
version	Show the Docker version information
info	Display system-wide information

Management Commands:

builder	Manage builds
buildx*	Docker Buildx
compose*	Docker Compose
container	Manage containers

```
PuTTY (inactive)
context      Manage contexts
image        Manage images
manifest     Manage Docker image manifests and manifest lists
network      Manage networks
plugin       Manage plugins
system       Manage Docker
trust        Manage trust on Docker images
volume       Manage volumes

Swarm Commands:
swarm        Manage Swarm

Commands:
attach       Attach local standard input, output, and error streams to a running container
commit       Create a new image from a container's changes
cp           Copy files/folders between a container and the local filesystem
create       Create a new container
diff         Inspect changes to files or directories on a container's filesystem
events       Get real time events from the server
export       Export a container's filesystem as a tar archive
history      Show the history of an image
import       Import the contents from a tarball to create a filesystem image
inspect      Return low-level information on Docker objects
kill         Kill one or more running containers
load         Load an image from a tar archive or STDIN
logs         Fetch the logs of a container
pause        Pause all processes within one or more containers
port         List port mappings or a specific mapping for the container
rename       Rename a container
restart      Restart one or more containers
rm           Remove one or more containers
rmi          Remove one or more images
save         Save one or more images to a tar archive (streamed to STDOUT by default)
start        Start one or more stopped containers
stats        Display a live stream of container(s) resource usage statistics
stop         Stop one or more running containers
tag          Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top          Display the running processes of a container
unpause      Unpause all processes within one or more containers
update       Update configuration of one or more containers
wait         Block until one or more containers stop, then print their exit codes

Global 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 DOCKER_HOST env var and default context set with
                     "docker context use")
-D, --debug           Enable debug mode
-H, --host list       Daemon socket to connect to
```



NIFTY
-0.24%



Q Search



ENG IN 12:56 PM 13-08-2024

```
PuTTY (inactive)
-D, --debug          Enable debug mode
-H, --host list       Daemon socket 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

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

root@myVM:~# docker container start web01
web01
root@myVM:~# docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAMES
5e360c47e659   nginx     "/docker-entrypoint..." 28 minutes ago Up 24 minutes  80/tcp
web01
root@myVM:~# docker container run --name server01 centos
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
root@myVM:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAMES
96c3164ddce7   centos    "/bin/bash"             33 seconds ago Exited (0) 31 seconds ago
server01
5e360c47e659   nginx     "/docker-entrypoint..." 30 minutes ago Up 26 minutes  80/tcp
web01
b6ad52270051   nginx     "/docker-entrypoint..." About an hour ago Created
trusting bouman
9a11591fb226   hello-world "/hello"                About an hour ago Exited (0) About an hour ago
charming_solomon
root@myVM:~# docker container run --name server02 -i -t centos
[root@0b0a85a8e71f /]# ps
PID TTY          TIME CMD
1 pts/0    00:00:00 bash
15 pts/0    00:00:00 ps
[root@0b0a85a8e71f /]# exit
exit
root@myVM:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
```

root@myVM: ~

```
2024-08-13T06:07:33.872989264Z      1 pts/0    00:00:00 bash
2024-08-13T06:07:33.872994364Z      15 pts/0    00:00:00 ps
2024-08-13T06:14:06.253877470Z [root@0dfcac480a26 /]# exit
2024-08-13T06:docker container prune -fdcac480a26 /]# root@myVM:~# Docker container prune -f
```

Deleted Containers:

```
0b0a85a8e71ffd08dabb8f6e48e08505981ba339802d742c9ec7464455580dbe
96c3164ddce75e43f37d6821da986ab2bc0bb184d86b46f39aa139392858ba91
b6ad522700512ed37215d75be6b8205550a8ef19bc29ef9ed7ee7a6c4b5941d9
9a11591fb226fb2646374f81d10fe959b773c1265a1746bec2e195fb32d31686
```

Total reclaimed space: 8B

root@myVM:~# docker container ls -a

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS	PORTS
fcd2949948f6	nginx	"/docker-entrypoint..."	5 hours ago	Up 5 hours	0.0.0.0:32768->80/tcp, :::32768->80/tcp
8fa9af116fa6	nginx	"/docker-entrypoint..."	5 hours ago	Up 5 hours	0.0.0.0:8080->80/tcp, :::8080->80/tcp
0dfcac480a26	centos	"/bin/bash" server003	5 hours ago	Up 5 hours	
5e360c47e659	nginx	"/docker-entrypoint..."	6 hours ago	Up 6 hours	80/tcp

root@myVM:~# docker container stop server003

server003

root@myVM:~# docker container rm server003

server003

root@myVM:~# docker container rm web01 web02 web03 -f

web01

web02

web03

root@myVM:~# docker container ls -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

root@myVM:~# docker image rm nginx centos

Untagged: nginx:latest

Untagged: nginx@sha256:6af79ae5de407283dcea8b00d5c37ace95441fd58a8b1d2aaled93f5511bb18c

Deleted: sha256:a72860cb95fd59e9c696c66441c64f18e66915fa26b249911e83c3854477ed9a

Deleted: sha256:1188c692bee9694c47db34046023dbd938d88f303f216ef689863741b2d1a900

Deleted: sha256:3eefccfd7e5fd8bbb2bd982509dc79206b056f22dd2b14553951a743833b0d09

Deleted: sha256:5234252bfd2bb1548a4998869e9a01aedfe3b319ce61acbe98f8aec223640e7

Deleted: sha256:b292d631e6ca5af8269dc2cf3ec47bel9faa0865b2aaa794daa2b8c25ea8cb4

Deleted: sha256:beda8840654459fe0efc1cd0bcae6a00b65b469cc999ebc41608b53c51fb93b4

Deleted: sha256:7a69d5090b2e7d873a365c81c590b2e6b87a702178b22b3c32c50d35eb7616fc

Deleted: sha256:e0781bc8667fb5ebf954df4ae52997f6f5568ec9f07e21e5db7c9d324ed41e1f

Untagged: centos:latest

Untagged: centos@sha256:a27fd8080b517143cbabb9dfb7c8571c40d67d534bbdee55bd6c473f432b177

Deleted: sha256:5d0da3dc976460b72c77d94c8alad043720b0416bfc16c52c45d4847e53fad6

Deleted: sha256:74ddd0ec08fa43d09f32636ba91a0a3053b02cb4627c35051aff89f853606b59

root@myVM:~#



Search



 root@myVM: ~

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS	PORTS
fcd2949948f6	nginx	"docker-entrypoint..." web03	5 hours ago	Up 5 hours	0.0.0.0:32768->80/tcp, :::32768->80/tcp
8fa9af116fa6	nginx	"docker-entrypoint..." web02	5 hours ago	Up 5 hours	0.0.0.0:8080->80/tcp, :::8080->80/tcp
0dfcac480a26	centos	"/bin/bash" server003	5 hours ago	Up 5 hours	
5e360c47e659	nginx	"/docker-entrypoint..." web01	6 hours ago	Up 6 hours	80/tcp

```
root@myVM:~# docker container stop server003
server003
```

```
root@myVM:~# docker container rm server003
server003
```

```
root@myVM:~# docker container rm web01 web02 web03 -f
```

web01

web02

web03

```
root@myVM:~# docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
root@myVM:~# docker image rm nginx centos
```

```
Untagged: nginx:latest
```

```
Untagged: nginx@sha256:6af79ae5de407283dcea8b00d5c37ace95441fd58a8b1d2aa1ed93f5511bb18c
```

Deleted: sha256:a72860cb95fd59e9c696c66441c64f18e66915fa26b249911e83c3854477ed9a

Deleted: sha256:1188c692bee9694c47db34046023dbd938d88f303f216ef689863741b2d1a900

Deleted: sha256:3eefccfd7e5fd8bbb2bd982509dc79206b056f22dd2b14553951a743833b0d09

Deleted: sha256:5234252bfd2bba1548a4998869e9a01aedfe3b319ce61acbe98f8aec223640e7

Deleted: sha256:b292d631e6ca5af8269dc2cf3ec47be1f9faa0865b2aaa794daa2b8c25ea8cb4

Deleted: sha256:beda8840654459fe0efc1cd0bcae6a00b65b469cc999ebc41608b53c51fb93b4

Deleted: sha256:7a69d5090b2e7d873a365c81c590b2e6b87a702178b22b3c32c50d35eb7616fc

Deleted: sha256:e0781bc8667fb5ebf954df4ae52997f6f5568ec9f07e21e5db7c9d324ed41e1f

```
Untagged: centos:latest
```

```
Untagged: centos@sha256:a27fd8080b517143cbbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
```

Deleted: sha256:5d0da3dc976460b72c77d94c8a1ad043720b0416bfc16c52c45d4847e53fadbb6

Deleted: sha256:74ddd0ec08fa43d09f32636ba91a0a3053b02cb4627c35051aff89f853606b59

```
root@myVM:~# docker container rm `docker container ls -a -q` -f
```

```
"docker container rm" requires at least 1 argument.
```

```
See 'docker container rm --help'.
```

```
Usage: docker container rm [OPTIONS] CONTAINER [CONTAINER...]
```