

A Complete Docker Command List in Cheat Sheet

Updated January 13, 2018  [CONTAINERS](#), [LINUX HOWTO](#)

The idea of using container was made possible by the namespaces feature added to Linux kernel version 2.6.24. Namespaces allows to create an isolated container that has no visibility or access to objects outside the container. LXC, LXD, systemd-nspawn, Linux-VServer, OpenVZ and Docker are some of the management tools for Linux containers.

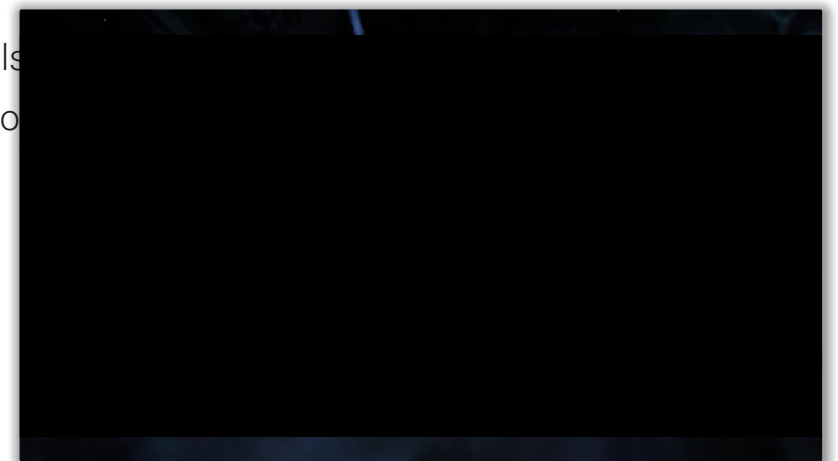
Main difference between native virtual machine and container is that in VM isolation is achieved by each VM have their own copies of Operating System files, libraries and application code. Whereas containers simply share the host operating system, including the kernel and libraries.

This article brings you docker command cheat list in a printable A4 size and also keep us posted if you need us to add more commands. Commands are categorized as follows:

Contents of Cheat sheet

PART 1

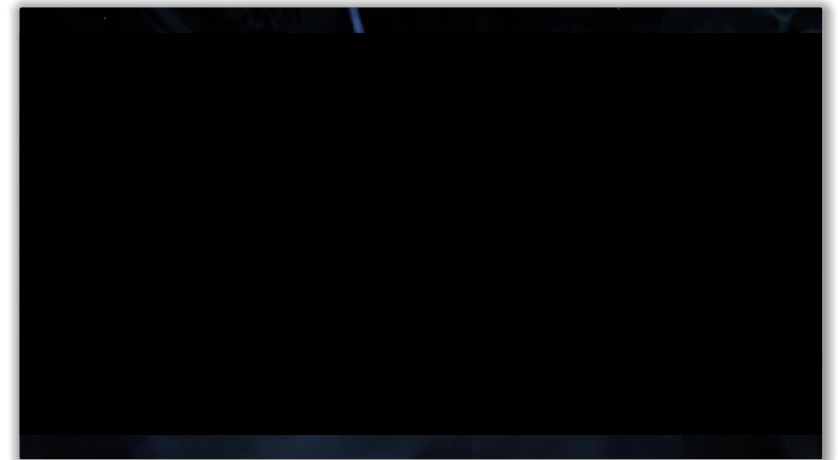
- 1. **Containers**



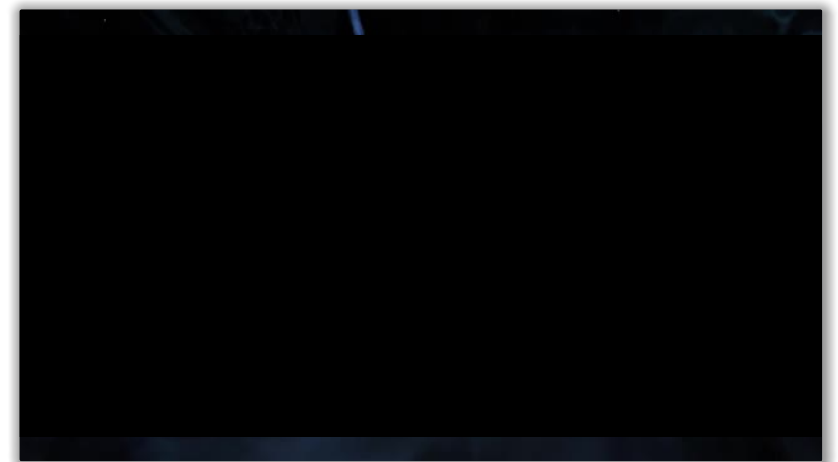
- 1.1. Lifecycle
- 1.2. Starting and Stopping
- 1.3. CPU Constraints
- 1.4. Memory Constraints
- 1.5. Capabilities
- 1.6. Info
- 1.7. Import / Export
- 1.8. Executing Commands
- 2. **Images**
 - 2.1. Lifecycle
 - 2.2. Info
 - 2.3. Cleaning up
 - 2.4. Load/Save image
 - 2.5. Import/Export container

PART 2

- 3. **Networks**
 - 3.1. Lifecycle
 - 3.2. Info
 - 3.3. Connection
- 4. **Registry & Repository**
- 5. **Volumes**
 - 5.1. Lifecycle
 - 5.2. Info



- 6. **Exposing ports**
- 7. **Tips**
 - 7.1. Get IP address
 - 7.2. Get port mapping
 - 7.3. Find containers by regular expression
 - 7.4. Get Environment Settings
 - 7.5. Kill running containers
 - 7.6. Delete old containers
 - 7.7. Delete stopped containers
 - 7.8. Delete dangling images
 - 7.9. Delete all images
 - 7.10. Delete dangling volumes



1. Containers

A lightweight virtual OS that run processes in full isolation.

1.1 Lifecycle

- `docker create` creates a container but does not start it.
- `docker rename` allows the container to be renamed.
- `docker run` creates and starts a container in one operation.
- `docker rm` deletes a container.
- `docker update` updates a container's resource limits.
 - `docker run --rm` : remove the container after it stops.
 - `docker run -v $HOSTDIR:$DOCKERDIR` : map the directory (\$HOSTDIR) on the host to a - docker container (\$DOCKERDIR).
 - `docker rm -v` : remove the volumes associated with the container.
 - `docker run --log-driver=syslog` : run docker with a custom log driver.

1.2 Starting and Stopping

- `docker start` starts a container so it is running.
- `docker stop` stops a running container.
- `docker restart` stops and starts a container.
- `docker pause` pauses a running container, "freezing" it in place.
- `docker unpause` will unpause a running container.
- `docker wait` blocks until running container stops.
- `docker kill` sends a SIGKILL to a running container.
- `docker attach` will connect to a running container.

1.3 CPU Constraints

CPU can be limited either using a percentage over all CPUs, or by using specific cores.

- `-c` or `cpu-shares`: 1024 means 100% of the CPU, so if we want the container to take 50%

➤ Give access to all devices:

- `docker run -it --privileged -v /dev/bus/usb:/dev/bus/usb debian bash`

1.6 Info

- `docker ps` shows running containers.
- `docker logs` gets logs from container. (You can use a custom log driver, but logs is only available for json-file and journald in 1.10).
- `docker inspect` looks at all the info on a container (including IP address).
- `docker events` gets events from container.
- `docker port` shows public facing port of container.
- `docker top` shows running processes in container.
- `docker stats` shows containers' resource usage statistics.
- `docker diff` shows changed files in the container's FS.

- `docker ps -a` shows all containers

1.7 Import / Export

- `docker cp` copy files from container and
- `docker export` export container as tarball archive

1.8 Executing Commands

`docker exec` to

2. Images

A template or blueprint for docker containers.

CPU, so if we want the container to take 50% of all CPU cores, we should specify 512 for instance, `docker run -ti --c 512 ...cpuset-cpus`
○: use only some CPU cores, for instance, `docker run -ti --cpuset-cpus=0,4,6 ...`

1.4 Memory Constraints

Memory can be limited using `-m` flag, for instance, `docker run -it -m 300M ubuntu:14.04 /bin/bash`

1.5 Capabilities

`cap-add` and `cap-drop`: Add or drop linux capabilities.

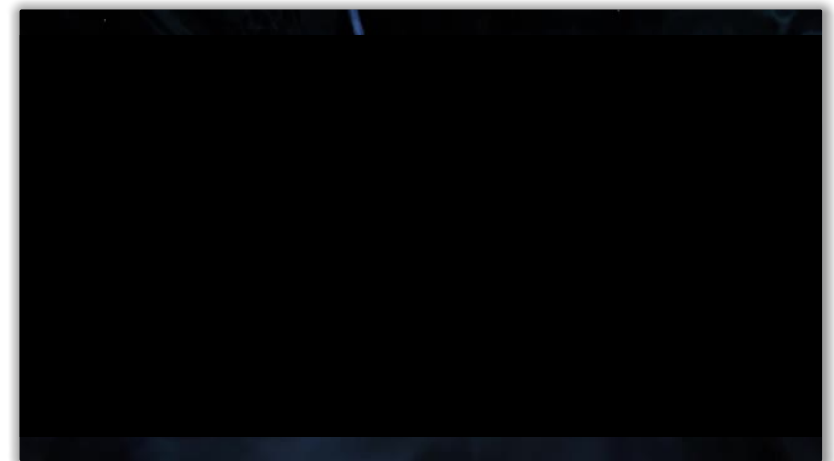
- Mount a FUSE based filesystem:
 - `docker run --rm -it --cap-add SYS_ADMIN --device /dev/fuse sshfs`
- Give access to a single device:
 - `docker run -it --device=/dev/ttyUSB0 debian bash`

A template or blueprint for docker containers.

2.1 Lifecycle

- `docker images` shows all images.
- `docker import` creates an image from a tarball.
- `docker build` creates image from Dockerfile.
- `docker commit` creates image from a container, pausing it temporarily if it is running.
- `docker rmi` removes an image.
- `docker load` loads an image from a tar archive as STDIN, including images and tags (as of 0.7).
- `docker save` saves an image to a tar archive stream to STDOUT with all parent layers, tags & versions (as of 0.7).

Linuxide - Linux Trends



2.2. Info

- `docker history` shows history of image.
- `docker tag` tags an image to a name (local or registry).

2.3. Cleaning up

- `docker rmi` remove specific images.
- `docker-gc` a tool to clean up images that are no longer used by any containers in a safe manner.

2.4. Load/Save image

- `docker load < my_image.tar.gz` load an image from file
- `docker save my_image:my_tag | gzip > my_image.tar.gz` save an existing image

2.5. Import/Export container

- `cat my_container.tar.gz | docker import - my_image:my_tag` import a container as an image from file
- `docker export my_container | gzip > my_container.tar.gz` export an existing container

3. Networks

A small def goes here

3.1. Lifecycle

- `docker network create`
- `docker network rm`

3.2. Info

- `docker network ls`
- `docker network inspect`

3.3. Connection

- `docker network connect`
- `docker network disconnect`

4. Registry & Repository

5. Volumes

Docker volumes are free-floating filesystems. They don't have to be connected to a particular container. You should use volumes mounted from data-only containers for portability.

5.1. Lifecycle

- `docker volume create`
- `docker volume rm`

5.2. Info

- `docker volume ls`
- `docker volume inspect`

6. Exposing ports

- `docker run -p 127.0.0.1:$HOSTPORT:$CONTAINER-PORT --name CONTAINER -t docker_image` mapping the container port to the host port using `-p`
- `EXPOSE <CONTAINERPORT>` expose port `CONTAINERPORT` at runtime (see `dockerfile`)
- `docker port CONTAINER $CONTAINERPORT` check the mapped port

7. Tips

7.1. Get IP address

```
> docker inspect some_docker_id | grep IPAddress  
| cut -d '"' -f 4  
or install jq:
```

```
> docker inspect  
workSettings.IP  
or using a go tool
```

```
> docker inspect  
<container_name>
```

7.2. Get port mapping

```
docker inspect  
workSettings.  
0).HostPort}} -
```

7.3. Find containers by

```
for i in $(dock  
TERN" | cut -f
```

7.4. Get Environment S

```
docker run --rm ubuntu env
```

A repository is a hosted collection of tagged images that together create the file system for a container.

A registry is a host -- a server that stores repositories and provides an HTTP API for managing the uploading and downloading of repositories.

Docker.com hosts its own index to a central registry which contains a large number of repositories.

- `docker login` to login to a registry.
- `docker logout` to logout from a registry.
- `docker search` searches registry for image.
- `docker pull` pulls an image from registry to local machine.
- `docker push` pushes an image to the registry from local machine.

7.5. Kill running containers

```
docker kill $(docker ps -q)
```

7.6. Delete old containers

```
docker ps -a | grep 'weeks ago' | awk '{print $1}' | xargs docker rm
```

7.7. Delete stopped containers

```
docker rm -v $(docker ps -a -q -f status=exited)
```

7.8. Delete dangling images

```
docker rmi $(docker images -q -f dangling=true)
```

7.9. Delete all images

```
docker rmi $(docker images -q)
```

7.10. Delete dangling volumes

```
docker volume rm $(docker volume ls -q -f dangling=true)
```

Linuxide - Linux Trends

You can download [docker commands cheat sheet pdf](#) , [Part 1](#) and [Part 2](#) image A4 format.

SHARE ON

[FACEBOOK](#)

[GOOGLE+](#)

[TWITTER](#)

[PINTEREST](#)

[LINKEDIN](#)

[REDDIT](#)

[Cheat Sheet](#), [Docker](#)

About Bobbin Zachariah

Founder of LinOxide, passionate lover of Linux and technology writer. Started his career in Linux / Opensource from 2000. Love traveling, blogging and listening music. Reach Bobbin Zachariah [about me](#) page and google plus page.



Like to become part of [Linoxide Team](#) and contribute tips? [Contact us here.](#)

Hand-picked related articles

```
himanshu@mylinuxbook-Inspiron-1525: ~
#include<stdio.h>
#include<unistd.h>

int main(void)
{
    printf("\n Welcome to Linoxide \n");

    while(1)
    {
        sleep(1);
    }

    return 0;
}
```

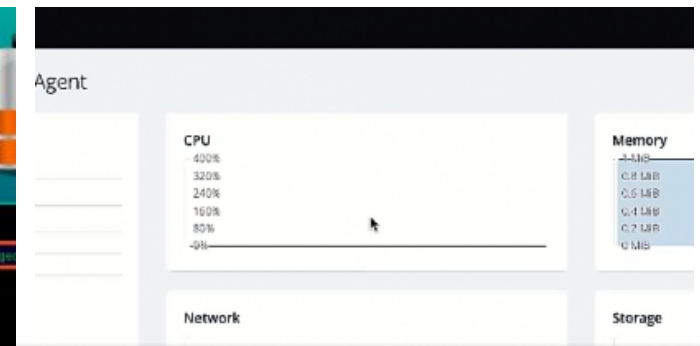
[A 5 Step Guide to Get Started with VIM Editor in Linux](#)

Linux offers powerful command line alternatives



[How to Check Laptop Battery Status from Linux Terminal](#)

What if you are too geek like me who wa



Comments

Your email address will not be published. Required fields are marked *

Name *

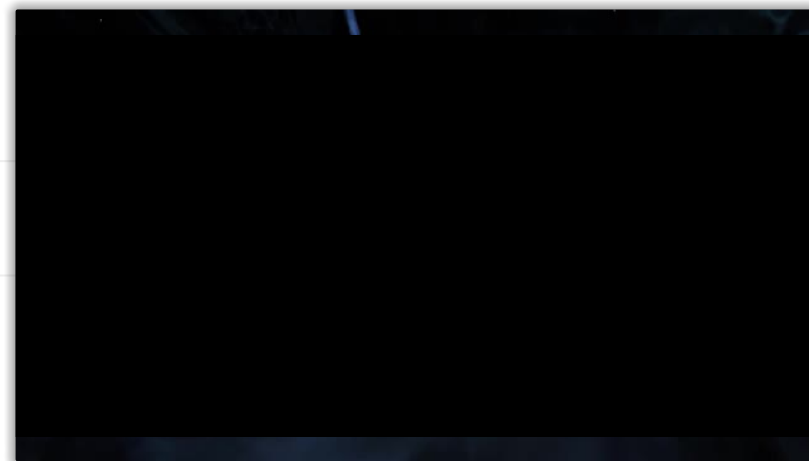
Email *

All comments are subject to moderation.

What is linuxide based on ? Windows or Linux ? Type your answer into the box

Submit Your Comment

Search Linoxide Tutorials



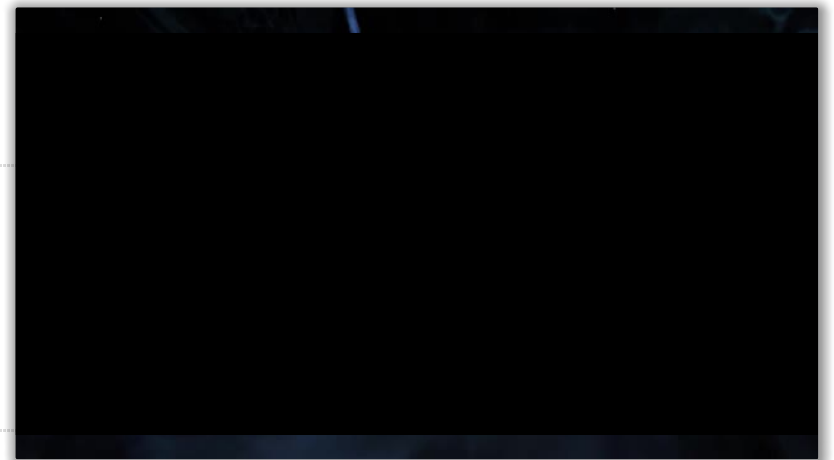
LATEST GIVEAWAYS!



How to Install Pip on CentOS 8



How to Install Cockpit on CentOS 8





How to Install CentOS 8 (Step by Step with Screenshots)

Linux Commands Cheat Sheet

[Download PDF Version](#)

Beginners Kubernetes Practical Guide

[Download Your Free Ebook](#)

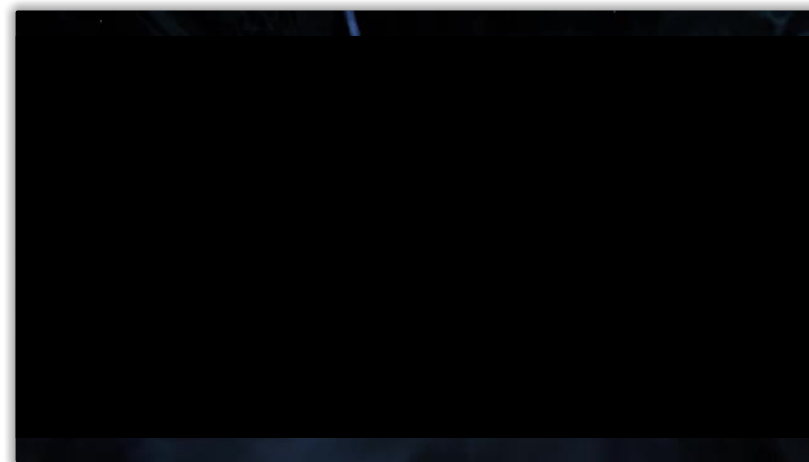
Enjoyed the Read?

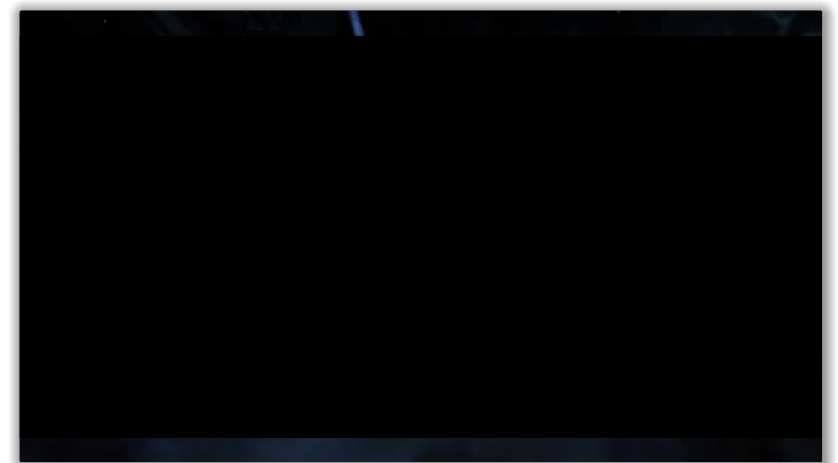
Don't miss our next article!

Enter Your Email *

SUBSCRIBE

100% privacy – We will never spam you







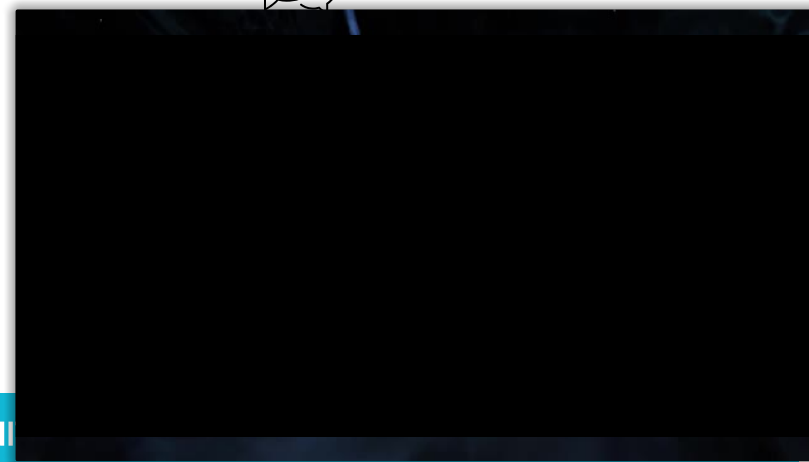
ABOUT & CONTACT PAGES



Notify Any Errors



Provide News & Tips



JOIN OUR 16.9K LINUX COMMUNITY



Ask & Share – [Join Our Facebook Community](#)

SUBSCRIBE TO FREE NEWSLETTER

Enjoyed the Read?

Don't miss our next article!

SUBSCRIBE

100% privacy – We will never spam you



[TERMS OF SERVICE](#) / [PRIVACY](#) / [CONTACT](#)

COPYRIGHT © 2019 BTREME. ALL RIGHTS RESERVED