

Docker Cheatsheet

This page presents a list of commonly used commands in docker.

Categories: [cheatsheet](#) [docker](#)

Access this page by simply typing in "**cs docker**" in your browser address bar if you have [bunnylol](#) configured.

Installation

Install Docker On Ubuntu [↗](#)

```
# Update the apt package index and install packages to allow apt to use a repository over HTTPS.
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release

# Add Docker's official GPG key.
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# set up the repository.
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install docker.
sudo apt update && sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y

# Verify if docker is successfully installed.
sudo docker run hello-world
```

[Copy](#)

Install Docker On Debian

```
# Update the apt package index and install packages to allow apt to use a repository over HTTPS.
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release

# Add Docker's official GPG key.
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Set up the repository.
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install docker.
sudo apt update && sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Verify if docker is successfully installed.
sudo docker run hello-world
```

[Copy](#)

Run Docker Commands Without Sudo

```
# Create a docker group and add yourself to it,  
# so that you don't need sudo before your docker command.  
# Note that you'll need to log out and log in for the changes to take effect.  
sudo groupadd docker  
sudo usermod -aG docker $USER
```

[Copy](#)

Commands

Image Management

```
# Build an image.  
docker build -t <image-name>  
  
# Build an Image without using cache.  
docker build -t <image-name> . -no-cache  
  
# List local images.  
docker images  
  
# Create a docker image from a running container.  
docker commit <container-name-or-id> <image-name>  
  
# Delete a local image.  
docker rmi <image-name>  
  
# Remove all unused images.  
docker image prune  
  
# Push docker image to docker registry.  
# Image name has to be in the form of <docker-registry-address>/<docker-repo>[:<docker-tag>].  
# Examples are:  
# * docker-repo.local/my-image  
# * docker-repo.local/my-image:debug  
docker push <image-name>  
  
# Remove dangling images.  
docker images -qf dangling=true | xargs docker rmi
```

[Copy](#)

Container Management

Copy

```
# Create a docker container.
docker create <image-name>

# Run a container from an image.
docker run <image-name>

# Run a container in interactive mode.
docker run -it <image-name>

# Run a container in the background.
docker run -d <image-name>

# Run a container with a port mapping.
docker run -p <host-port>:<container-port> <image-name>

# Start an existing container.
docker start <container-id-or-name>

# Restart a container.
docker restart <container-id-or-name>

# Pause a container.
docker pause <container-id-or-name>

# Unpause a container.
docker unpause <container-id-or-name>

# Wait until running container stops.
docker wait <container-id-or-name>

# Stop a container.
docker stop <container-id-or-name>

# Kill a container.
docker kill <container-id-or-name>

# Attach a container. (C-p C-q to detach)
docker attach <container-id-or-name>

# Rename a container.
docker rename <container-id-or-name> <new-name>

# Update a container.
docker update [OPTIONS] <container-id-or-name>

# Remove a stopped container.
docker rm <container-id-or-name>

# Follow the logs of a container.
docker logs -f <container-id-or-name>

# Inspect a running container.
docker inspect <container-id-or-name>

# List running containers.
docker ps

# List all containers.
docker ps -a
```

File Copy

```
# Copy files from host to container.
docker cp <host-filepath> <container-id-or-name>:<container-filepath>

# Copy files from container to host.
docker cp <container-id-or-name>:<container-filepath> <host-filepath>

# Export all files in container to host.
docker export <container-id-or-name> -o <filename>
```

Copy

Network Management

```
# Create a docker network.
docker network create --subnet <subnet> --gateway <gateway> <docker-network-name>

# Remove a docker network.
docker network rm <docker-network-name>

# List all docker networks.
docker network ls

# Inspect a docker network.
docker network inspect <docker-network-name>

# Connect a container with a docker network.
docker network connect <docker-network-name> <container-id-or-name>

# Disconnect a container with a docker network.
docker network disconnect <docker-network-name> <container-id-or-name>
```

Copy

Docker In Docker

```
# Run a docker container which maps the /var/run/docker.sock file,
# so that you can run another docker container within this one.
docker run -v /var/run/docker.sock:/var/run/docker.sock -it <image-name>
```

Copy

Dockerfile Builder Commands

Command	Description
FROM <image-name>	Docker base image name.
MAINTAINER <email>	Maintainer's email address.
COPY <src-file-on-host> <dst-file-on-docker>	Copy files from host to docker image.
RUN <command>	Run a command when building the image.
USER <username>	Set the default user name.
WORKDIR <dir>	Set the default working directory.
CMD <command>	Start up default command which can be overridden by docker cli.
ENTRYPOINT <command>	Start up default command which cannot be overridden by docker cli.
ENV <env-name> <env-value>	Environment variables.

Docker Config

Insecure Docker Registry

```
# Add the following content to /etc/docker/daemon.json if you using an insecure docker registry.
```

[Copy](#)

```
{  
  "insecure-registries" : ["<your-docker-register-server-ip-or-hostname>"]  
}
```