

# Docker CheatSheet

# KUBERNETES

- PDF Link: [cheatsheet-docker-A4.pdf](#), Category: tools
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-docker-A4>
- Related posts: Kubernetes Yaml, [#denny-cheatsheets](#)

File me Issues or star this repo.

## 1.1 Docker start service

Name	Summary
Start a ubuntu test env	<code>docker run ubuntu:16.04 /bin/echo hello world</code>
Start a centos test env	<code>docker run centos:centos6 /bin/echo hello world</code>
Start nginx	<code>docker run -t -d -p 8080:80 --name nginx-test nginx</code>
Start a sshd server	<code>docker run -t -d --privileged -p 5022:22 denny/sshd:latest /usr/sbin/sshd -D</code>
Start a jenkins server	<code>docker run -p 8080:8080 denny/jenkins:1.0</code>
Start a ftp server	<code>docker run -t -d -p 21:21 -p 20:20 -e USERNAME=\${username} -e PASSWORD=\${password} den</code>

## 1.2 Docker Check Status

Name	Summary
Tail container logs	<code>docker logs --tail 5 \$container_name</code>
Check container healthcheck status	<code>docker inspect --format '{{.State.Health}}' \$container_name</code>
List all containers	<code>docker ps, docker ps -a</code>
List containers by labels	<code>docker ps --filter "label=org.label-schema.group"</code>
List all images	<code>docker images, docker images -a</code>

## 1.3 Docker Cleanup

Name	Summary
Remove the specified image	<code>docker rmi &lt;imagename&gt;</code>
Remove all docker images	<code>docker rmi \$(docker images -q)</code>
Remove orphaned docker volumes	<code>docker volume rm \$(docker volume ls -qf dangling=true)</code>
Remove dead containers	<code>docker rm \$(docker ps --filter status=dead -qa)</code>

## 1.4 Container Basic

Name	Summary
Start docker container	<code>docker run -p 4000:80 imgname</code>
Start docker container in detached mode	<code>docker run -d -p 4000:80 imgname</code>
Enter a running container	<code>docker exec -it [container-id] sh</code>
Stop container	<code>docker stop &lt;hash&gt;</code>
Remove container	<code>docker rm &lt;hash&gt;</code>
Remove all containers	<code>docker rm \$(docker ps -a -q)</code>
Force shutdown of one given container	<code>docker kill &lt;hash&gt;</code>
Login to docker hub	<code>docker login</code>
Tag <image>	<code>docker tag &lt;image&gt; username/repo:tag</code>
Docker push a tagged image to repo	<code>docker push username/repo:tag</code>
Run image from a given tag	<code>docker run username/repo:tag</code>
Create docker image	<code>docker build -t denny/image:test .</code>
Docker files	<code>/var/lib/docker, /var/lib/docker/devicemapper/mnt</code>

## 1.5 Docker Compose

Name	Summary
Change entrypoint to run nothing	<code>entrypoint: ["tail", "-f", "/dev/null"]</code>
Change restart policy	<code>restart: always</code> , Link: <a href="#">Compose file version 3 reference</a>
Mount file as volume	<code>\$PWD/httpd/httpd.conf:/usr/local/apache2/conf/httpd.conf:ro</code> <code>sample-mount-file.yml</code>
Start compose env	<code>docker-compose up</code> , <code>docker-compose up -d</code>
Stop compose env	<code>docker-compose down</code> , <code>docker-compose down -v</code>
Check logs	<code>docker-compose logs</code>

## 1.6 Ubuntu docker: Install missing packages

Name	Summary
Pull ubuntu docker image	<code>docker pull ubuntu</code>
man: command not found	<code>apt-get update</code> , <code>apt-get install man</code>
ping: command not found	<code>apt-get update</code> , <code>apt-get install iputils-ping</code>
dig: command not found	<code>apt-get install dnsutils</code>

## 1.7 Docker Socket file

Name	Summary
Links	Link: <a href="#">Develop with Docker Engine SDKs and API</a>
Run container mounting socket file	<code>docker run -v /var/run/docker.sock:/var/run/docker.sock -ti alpine sh</code>
A different docker socket file	<code>export DOCKER_HOST=unix:///my/docker.sock</code>
List containers	<code>curl -XGET --unix-socket /var/run/docker.sock http://localhost/containers/json</code>
Stop container	<code>curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/&lt;id&gt;/stop</code>
Start container	<code>curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/&lt;id&gt;/start</code>
List events	<code>curl --unix-socket /var/run/docker.sock http://localhost/events</code>
Create container	<code>curl -XPOST --unix-socket /var/run/docker.sock -d '{"Image":"nginx:alpine"}'</code>

## 1.8 Container Runtime

Name	Summary
dockerd	
rkt	a pod-native container engine for Linux.
CRI-O	Open Container Initiative-based implementation of Kubernetes Container Runtime Interface

## 1.9 Resource Reference

Name	Summary
Docker SDK	<a href="https://docs.docker.com/develop/sdk/examples/">https://docs.docker.com/develop/sdk/examples/</a>
Docker REST API	<a href="https://docs.docker.com/engine/api/v1.27/#tag/Container">https://docs.docker.com/engine/api/v1.27/#tag/Container</a>
Docker Hub auto build	<a href="https://docs.docker.com/docker-hub/builds/#build-statuses-explained">https://docs.docker.com/docker-hub/builds/#build-statuses-explained</a>

## 1.10 More Resources

License: Code is licensed under MIT License.