

Docker CheatSheet

CLOUD

- PDF Link: [cheatsheet-docker-A4.pdf](#), Category: Cloud
- 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 Trouble Shooting

Name	Summary
Docker push: manifest invalid	Re-push a new version of the same docker tag may fail, due to permis
Docker pull: missing signature key	Docker push again to resolve the issue
Docker cp: Error response from daemon: not a directory	container folder is in a symbol link
Find process id by container name	<code>docker top \$container_id</code> , or <code>docker top \$container_name</code>
List resource usage by containers	<code>docker stats</code>
Get dockerd storage driver	<code>docker info</code> , then check Storage Driver
docker-containerd-shim	The Docker four components: Docker engine, containerd, contain

1.2 Docker Basic

Name	Summary
Install docker on Ubuntu	<code>apt-get install docker.io</code>
Install docker on CentOS	Use docker repo https://download.docker.com/linux/centos/docker-ce.repo
Install docker in Debian 10	Link: How To Install and Use Docker on Debian 10
Install old docker version	GitHub: install-old-docker.md

1.3 Docker start service

Name	Summary
Start a ubuntu test env	<code>docker run ubuntu:16.04 /bin/echo hello world</code>
Start a ubuntu 18.04 test env	<code>docker run ubuntu:18.04 /bin/echo hello world</code>
Start a container run and stop	<code>docker run --rm ubuntu:18.04 /bin/echo hello world</code>
Start a debian9 test env	<code>docker run debian:9 /bin/echo hello world</code>
Start a centos test env	<code>docker run centos:centos6 /bin/echo hello world</code>
Start a jenkins server	<code>docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts</code>
Start a nginx server	<code>docker run -t -d -p 8080:80 --name nginx-test nginx</code>
Start a mysql server	<code>docker run -e MYSQL_ROOT_PASSWORD=password123 -e MYSQL_DATABASE=wordpress -d mys</code>
Start a nexus server	<code>docker run -d -p 8082:8081 --name nexus -v /data/nexus-data:/nexus-data sonatype</code>
Start a sshd server	<code>docker run -t -d --privileged -p 5022:22 denny/sshd:latest /usr/sbin/sshd -D</code>
Start a ftp server	<code>docker run -t -d -p 21:21 -p 20:20 -e USERNAME=\${username} -e PASSWORD=\${passwor</code>

1.4 Container Runtime

Name	Summary
dockerd	
containerd	
cri-o	From Redhat
rkt	a pod-native container engine for Linux from CoreOS. Stopped maintainance now
Amazon ACS	supports DC/OS, Swarm, Kubernetes
CoreOS Fleet	
Cloud Foundry Diego	Not actively maintained any more
Reference	CheatSheet: Docker, CheatSheet: CRI-O, CheatSheet: rkt, CheatSheet: containerd

1.5 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>
Start container with entrypoint changed	<code>docker run -t -d --entrypoint=/bin/sh "\$docker_image"</code>
Enter a running container	<code>docker exec -it <container-id> sh</code>
Upload local file to container filesystem	<code>docker cp /tmp/foo.txt mycontainer:/foo.txt</code>
Download container file local filesystem	<code>docker cp mycontainer:/foo.txt /tmp/foo.txt</code>
Stop container	<code>docker stop <hash></code>
Remove container	<code>docker rm <hash></code>
Remove all containers	<code>docker rm \$(docker ps -a -q)</code>
Force shutdown of one given container	<code>docker kill <hash></code>
Login to docker hub	<code>docker login</code>
Tag <image>	<code>docker tag <image> 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>

1.6 Docker Cleanup

Name	Summary
Remove unused docker images	<code>delete-unused-images.sh</code>
Delete all containers	<code>delete-all-containers.sh</code>
Remove exited containers	<code>docker rm \$(docker ps --filter status=exited -qa)</code>
Docker prune images	<code>docker image prune -f</code>
Docker prune volumes	<code>docker volume prune -f</code>
Remove the specified image	<code>docker rmi <imagename></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>

1.7 Dockerfile

Name	Summary
Change entrypoint to run nothing	<code>entrypoint: ["tail", "-f", "/dev/null"]</code>
Set timezone in Dockerfile	<code>RUN ln -snf /usr/share/zoneinfo/\$TZ /etc/localtime && echo \$TZ > /etc/timezone</code>
Define multiple line command	GitHub: Dockerfile-example-multiline

1.8 Docker Compose

Name	Summary
Change restart policy	<code>restart: always</code> , Link: Compose file version 3 reference
Mount file as volume	<code>\$PWD/httpd/httpd.conf:/usr/local/apache2/conf/httpd.conf:ro</code> GitHub: sample-mount-file.yml
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>

1.9 Docker Containers

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>
Start docker container and remove when exit	<code>docker run -rm -it <imgname> sh</code>
Enter a running container	<code>docker exec -it [container-id] sh</code>
Stop container	<code>docker stop <hash></code>
List all containers	<code>docker ps, docker ps -a</code>
Remove container	<code>docker rm <hash>, docker rm \$(docker ps -a -q)</code>
Force shutdown of one given container	<code>docker kill <hash></code>
Login to docker hub	<code>docker login</code>
Run image from a given tag	<code>docker run username/repo:tag</code>
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 containers by labels	<code>docker ps --filter "label=org.label-schema.group"</code>

1.10 Docker Images

Name	Summary
List all images	<code>docker images, docker images -a</code>
Create docker image	<code>docker build -t denny/image:<tag> .</code>
Docker push a tagged image to repo	<code>docker push denny/image:<tag></code>
Show the history of an image	<code>docker history <image_name></code>
Export image to file	<code>docker save <image_name> > my_img.tar</code>
Load image to local registry	<code>docker load -i my_img.tar</code>
Tag <image>	<code>docker tag <image> username/repo:tag</code>

1.11 Docker Socket file

Name	Summary
Run container mounting socket file	<code>docker run -v /var/run/docker.sock:/var/run/docker.sock -it 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/</code>
Stop container	<code>curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/<id></code>
Start container	<code>curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/<id></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>
Links	Link: Develop with Docker Engine SDKs and API

1.12 Docker Conf

Name	Summary
Docker files	<code>/var/lib/docker, /var/lib/docker/devicemapper/mnt</code>
Docker for Mac	<code>~/Library/Containers/com.docker.docker/Data/</code>

1.13 Ubuntu docker: Install missing packages

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

1.14 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 containers	<code>docker ps</code>
List all containers	<code>docker ps -a</code>
List containers by labels	<code>docker ps --filter "label=org.label-schema.group"</code>
List all images	<code>docker images -a</code>

1.15 Resource Reference

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

1.16 More Resources

License: Code is licensed under MIT License.