

Docker CheatSheet

KUBERNETES

- PDF Link: [cheatsheet-docker-A4.pdf](#)
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-docker-A4>
- Category: tools

File me Issues or star this repo.

See more CheatSheets from Denny: [#denny-cheatsheets](#)

1.1 Docker Compose

Name	Summary
Change entrypoint to run nothing	<code>entrypoint: ["tail", "-f", "/dev/null"]</code>
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> samples/mount-file.y
Start compose env	<code>docker-compose up, docker-compose up -d</code>
Stop compose env	<code>docker-compose down, docker-compose down -v</code>
Check logs	<code>docker-compose logs</code>

1.2 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>

1.3 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 <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>

1.4 Docker Socket file

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

1.5 Cleanup

Name	Summary
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>

1.6 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

1.7 Scripts

- Docker build

```
cd "$dir" && cat "$dockerfile" | docker build -t "$docker_image" -f - .
```

- Use google docker registry

```
docker build -t gcr.io/<MY_REPO>/<MY_IMAGE>:<MY_TAG> .
gcloud docker -- push gcr.io/<MY_REPO>/<MY_IMAGE>:<MY_TAG>
```

- Delete all containers

```
delete-all-containers.sh
```

```
curl -L https://raw.githubusercontent.com/dennyzhang/cheatsheet-docker-A4/master/delete-all-containers.sh | b
```

- Install test kit

```
container-install-devkit.sh
```

```
apt-get -y update
apt-get install -y curl netcat
```

```
curl -L https://raw.githubusercontent.com/dennyzhang/cheatsheet-docker-A4/master/container-install-devkit.sh
```

- Clean up Disk

Remove All Useless Resources.

```
docker ps --filter status=exited -aq | xargs -r docker rm -v
```

Remove unused docker images

```
docker rmi $(docker images | grep "<none>" | awk -F' ' '{print $3}')
```

Remove intermediate containers generated during docker build

```
docker ps -a | grep "/bin/sh -c" | awk -F' ' '{print $1}' | xargs docker rm
```

Remove Image with <none> string

```
echo "Remove docker images with <none> string"
if docker images | grep none | tee; then
  docker rmi $(docker images | grep "<none>" | awk -F' ' '{print $3}') | tee
fi
```

1.8 More Resources

License: Code is licensed under MIT License.