QuickRef.ME

Docker

This is a quick reference cheat sheet for Docker. And you can find the most common Docker commands here.

# Getting started

### Getting started

Create and run a container in background

```
$ docker run -d -p 80:80 docker/getting-started
```

- `-d` - Run the container in detached mode
- `-p 80:80` - Map port 80 to port 80 in the container
- `docker/getting-started` - The image to use

Create and run a container in foreground

```
$ docker run -it -p 8001:8080 --name my-nginx nginx
```

- `-it` - Interactive bash mode
- `-p 8001:8080` - Map port 8001 to port 8080 in the container
- `--name my-nginx` - Specify a name
- `nginx` - The image to use

General commands

`docker ps`	List running containers
`docker ps -a`	List all containers
`docker ps -s`	List running containers (with CPU / memory)

`docker images`	List all images
`docker exec -it <container> bash`	Connecting to container
`docker logs <container>`	Shows container's console log
`docker stop <container>`	Stop a container
`docker restart <container>`	Restart a container
`docker rm <container>`	Remove a container
`docker port <container>`	Shows container's port mapping
`docker top <container>`	List processes
`docker kill <container>`	Kill a container

Parameter `<container>` can be container id or name

# Containers

Starting & Stopping

`docker start nginx-server`	Starting
`docker stop nginx-server`	Stopping
`docker restart nginx-server`	Restarting
`docker pause nginx-server`	Pausing
`docker unpause nginx-server`	Unpausing
`docker wait nginx-server`	Blocking a Container
`docker kill nginx-server`	Sending a SIGKILL
`docker attach nginx-server`	Connecting to an Existing Container

Information

`docker ps`	List running containers
`docker ps -a`	List all containers
`docker logs nginx-server`	Container Logs

`docker inspect nginx-server`	Inspecting Containers
`docker events nginx-server`	Containers Events
`docker port nginx-server`	Public Ports
`docker top nginx-server`	Running Processes
`docker stats nginx-server`	Container Resource Usage
`docker diff nginx-server`	Lists the changes made to a container.

Creating

```
docker create [options] IMAGE
  -a, --attach               # attach stdout/err
  -i, --interactive          # attach stdin (interactive)
  -t, --tty                  # pseudo-tty
      --name NAME            # name your image
  -p, --publish 5000:5000    # port map (host:container)
      --expose 5432          # expose a port to containers
  -P, --publish-all          # publish all ports
      --link container:alias # linking
  -v, --volume `pwd`:/app    # mount (absolute paths needed)
  -e, --env NAME=hello       # env vars
```

Example

```
$ docker create --name my_redis --expose 6379 redis:3.0.2
```

Manipulating

Renaming a Container

```
docker rename my-nginx nginx-server
```

Removing a Container

```
docker rm nginx-server
```

Updating a Container

```
docker update --cpu-shares 512 -m 300M nginx-server
```

# Images

Manipulating

`docker images`	Listing images
`docker rmi nginx`	Removing an image
`docker load < ubuntu.tar.gz`	Loading a tarred repository
`docker load --input ubuntu.tar`	Loading a tarred repository
`docker save busybox > ubuntu.tar`	Save an image to a tar archive
`docker history`	Showing the history of an image
`docker commit nginx`	Save a container as an image.
`docker tag nginx eon01/nginx`	Tagging an image
`docker push eon01/nginx`	Pushing an image

Building Images

```
$ docker build .
$ docker build github.com/creack/docker-firefox
$ docker build - < Dockerfile
$ docker build - < context.tar.gz
$ docker build -t eon/nginx-server .
$ docker build -f myOtherDockerfile .
$ curl example.com/remote/Dockerfile | docker build -f - .
```

# Networking

Manipulating

Removing a network

```
docker network rm MyOverlayNetwork
```

Listing networks

```
docker network ls
```

Getting information about a network

```
docker network inspect MyOverlayNetwork
```

Connecting a running container to a network

```
docker network connect MyOverlayNetwork nginx
```

Connecting a container to a network when it starts

```
docker run -it -d --network=MyOverlayNetwork nginx
```

Disconnecting a container from a network

```
docker network disconnect MyOverlayNetwork nginx
```

Creating Networks

```
docker network create -d overlay MyOverlayNetwork

docker network create -d bridge MyBridgeNetwork

docker network create -d overlay \
  --subnet=192.168.0.0/16 \
  --subnet=192.170.0.0/16 \
  --gateway=192.168.0.100 \
  --gateway=192.170.0.100 \
  --ip-range=192.168.1.0/24 \
  --aux-address="my-router=192.168.1.5" \
  --aux-address="my-switch=192.168.1.6" \
  --aux-address="my-printer=192.170.1.5" \
  --aux-address="my-nas=192.170.1.6" \
  MyOverlayNetwork
```

# Miscellaneous

Docker Hub

`docker search search_word`	Search docker hub for images.
`docker pull user/image`	Downloads an image from docker hub

`docker pull user/image`	Downloads an image from docker hub.
`docker login`	Authenticate to docker hub
`docker push user/image`	Uploads an image to docker hub.

Registry commands

Login to a Registry

```
$ docker login
$ docker login localhost:8080
```

Logout from a Registry

```
$ docker logout
$ docker logout localhost:8080
```

Searching an Image

```
$ docker search nginx
$ docker search nginx --stars=3 --no-trunc busybox
```

Pulling an Image

```
$ docker pull nginx
$ docker pull eon01/nginx localhost:5000/myadmin/nginx
```

Pushing an Image

```
$ docker push eon01/nginx
$ docker push eon01/nginx localhost:5000/myadmin/nginx
```

Batch clean

`docker stop -f $(docker ps -a -q)`	Stopping all containers
`docker rm -f $(docker ps -a -q)`	Removing all containers
`docker rmi -f $(docker images -q)`	Removing all images

Volumes

Check volumes

```
$ docker volume ls
```

Cleanup unused volumes

```
$ docker volume prune
```

POPULAR

Bash scripting
Vim
RegEX
QuickRef

RECENT

C++
Java
Tmux
Lsof

WEBSITE

Privacy Policy
About