

# Docker 1.13 Cheat Sheet

David McKay, Author

## Building an Image

Building an image with the name "acme-corp/project" and the tag "latest"

```
$ docker image build -t acme-corp/project:latest
```

## Managing Images

### Search Docker Hub

```
$ docker search node
$ docker search -f is-official=true node # Show only
official images
```

### Pull an image

```
$ docker image pull alpine:3.5
```

### Delete an image

```
$ docker image rm alpine:3.5
```

## Working with Containers

### Listing running containers

```
$ docker container ls
```

### Listing running containers

```
$ docker container ls
```

### Execute command in an already running container

```
$ docker container exec (container name | id) echo
"Hello, world"
$ docker container exec -it (container name | id) sh
```

*Tip: you can usually get away with only the first two characters of the container identifier, provided there are no collisions.*

### Get log output from a container

```
$ docker container logs (container name | id)
```

### Follow log output from a container

```
$ docker container logs -f (container name | id)
```

## About the Author

David is a technical lead, developer, trainer, speaker, software and technology consultant, and a practicing Stoic who likes containers.

# Docker 1.13 Cheat Sheet

David McKay, Author

## Running Containers

Running **hello-world:latest** container

```
$ docker container run --name mycontainer hello-world:latest
```

*Warning: naming containers is generally not a good idea*

Run **hello-world:latest** container and delete after exit

```
$ docker container run --rm hello-world:latest
```

*Warning: using **--rm** will remove the container when it exists, successfully or on error*

Running a container and specifying a command to run

```
$ docker container run alpine:3.5 echo "Hello, world!"
```

Run a container and connect to terminal

```
$ docker container run -it alpine:3.5 sh
```

Run a container and publish container port 80 on host port 10000

```
$ docker container run -p 10000:80 nginx:latest
```

Run a container and publish exposed ports on random host port (>=32768)

```
$ docker container run -P nginx:latest
```

Run a container in the background

```
$ docker container run -d nginx:latest
```

## Running Containers (continued)

Run a container and mount a host directory

```
$ docker container run -v /home/
rawkode/code/project:/code php:7-cli
```

*Warning: Docker requires absolute paths when mounting a local directory into a container*

*Tip: if you're in the same directory when executing the command, use \$PWD*

Run a container and specify a restart policy for the container (the default is none)

```
$ docker container run --restart=on-failure:5
nginx:latest # Restart 5 times on failure
$ docker container run --restart=always nginx:latest
# Always restart, regardless of exit code
```

## Maintenance

Check Docker disk usage

```
$ docker system df
```

Removing a container by name or id

```
$ docker container rm (container_name | id)
$ docker container rm -f (container_name | id) # -f /
--force will stop and delete the container
$ docker container rm -v (container_name | id) # -v /
--volumes will also remove it's volumes
```

Remove stopped containers, unused volumes, unused networks and danling images

```
$ docker system prune
```