

Container Running

Run a new container

```
docker container run [options] IMAGE[:TAG]
[COMMAND]
```

Options

`--rm` remove container on exit

`--name` add container ID

`-e` set environment variable

`-p` publish container ports to host

`--network` connect container to network

`--net-alias` network scoped alias for the container

`-d` container in background / detached

`-it` run interactively

`-v=[host-src:]container-dest[:<options>]` Volume

`-v $(pwd):/app` map the pwd into the 'app' folder

Start an existing Container

```
docker container start [container]
```

`-a` attach

`-i` interactive

Stop a running container

```
docker container stop [container]
```

Kill a running container

```
docker container kill [container]
```

Container Management Commands

List containers

```
docker container ls
```

`docker ps` old command

Create a container

```
docker container create [OPTIONS]
IMAGE
```

Remove a container

```
docker container rm
```

`-f` force removal of running container

Remove all stopped containers

```
docker container prune
```

List processes in specific container

```
docker container top [container]
```

Show logs on specific container

```
docker container logs [container]
```

List docker container commands

```
docker container
```

Container commands

start a new container interactively

```
docker container run -it
```

Run additional process in running container

```
docker container exec CONTAINER
[CMD]
```

```
docker container exec -it
my_alpine ping my_ubuntu
```

```
docker container run --rm --net
bcs alpine nslookup search
```

List of ports forwarding traffic from host to container

```
docker container port [container]
```

Display detailed container information

```
docker container inspect
```

Management

What version

```
docker version
```

Configuration values for the docker engine

```
docker info
```

List of all commands

```
docker
```

Networks

List network

```
docker network ls
```

Inspect network

```
docker network inspect
```

Create a network

```
docker network create [network]
```

Attach a network to container

```
docker network connect [network]
[container]
```

Detach a network from container

```
docker network disconnect [network]
[container]
```

Images

List images

```
docker images [OPTIONS]
```

Re-tag an image

```
docker image tag SOURCE_IMAGE[:tag]
TARGET_IMAGE[:tag]
```

Log in to registry (the Docker Hub by default)

```
docker login
```

Push an image to the registry

```
docker push [options] NAME[:TAG]
```

Pull an image

```
docker pull [options] NAME[:TAG]
```

Build an image from a Dockerfile

```
docker image build [OPTIONS] PATH
```

`-t` tag



Docker File - image recipies

Set the base image for further instructions

```
FROM <image>[:<tag>]
```

Set environment variables

```
ENV <key>=<value>
```

Set instruction to execute

```
RUN <command> [&& <command>]
```

Set the ports that the container listens on

```
EXPOSE <port> [<port>/<protocol>...]
```

Default command for executing container

```
CMD ["executable", "param1", "param2"]
```

Working directory changed for later instructions

```
WORKDIR /path/to/workdir
```

Copy files

```
COPY [--chown=<user>:<group>] <src>...  
<dest>
```

Docker Compose

Start system

```
docker-compose up
```

```
--build - rebuild container
```

```
-d detach/launch in background
```

Stop and Remove system

```
docker-compose down
```

List containers

```
docker-compose ps
```

Docker Volume

Where does the container keep the volume?

```
docker container inspect [options] CONTAINER
```

Detailed information on one or more volume

```
docker volume inspect [OPTIONS] VOLUME
```

List volumes

```
docker volume ls [OPTIONS]
```

Remove all unused local volumes

```
docker volume prune
```

