

Container Running	Container Management Commands	Management
Run a new container	List containers	What version
<code>docker container run [options] IMAGE[:TAG] [COMMAND]</code>	<code>docker container ls</code>	<code>docker version</code>
Options	<code>docker ps old command</code>	Configuration values for the docker engine
<code>--rm</code>	Create a container	<code>docker info</code>
	<code>docker container create [OPTIONS] IMAGE</code>	List of all commands
	Remove a container	<code>docker</code>
<code>--name</code>	<code>docker container rm</code>	
	<code>-f</code> force removal of running container	Networks
<code>-e</code>	Remove all stopped containers	List network
	<code>docker container prune</code>	<code>docker network ls</code>
	List processes in specific container	Inspect network
<code>-p</code>	<code>docker container top [container]</code>	<code>docker network inspect</code>
	Show logs on specific container	Create a network
	<code>docker container logs [container]</code>	<code>docker network create [network]</code>
<code>--network</code>	List docker container commands	Attach a network to container
	<code>docker container</code>	<code>docker network connect [network] [</code>
<code>--net-alias</code>	Container commands	Detach a network from container
	start a new container interactively	<code>docker network disconnect [network]</code>
	<code>docker container run -it</code>	
<code>-d</code>	Run additional process in running container	Images
	<code>docker container exec CONTAINER [CMD]</code>	List images
	<code>docker container exec -it my_alpine</code>	<code>docker images [OPTIONS]</code>
<code>-it</code>	run interactively	ping my_ubuntu
	List of ports forwarding traffic from host to container	Re-tag an image
<code>-v=[host-path] : [container-path] : [options]</code>	Volume	<code>alpine nslookup search</code>
	Display detailed container information	<code>docker image tag SOURCE_IMAGE[:tag]</code>
<code>-v \$(pwd) : /app</code>	map the	Log in to registry (the Docker Hub by default)
	<code>docker container inspect</code>	<code>docker login</code>
	the '/app' folder	Push an image to the registry
Start an existing Container		<code>docker push [options] NAME[:TAG]</code>
<code>docker container start [container]</code>		Pull an image
<code>-a attach</code>		<code>docker pull [options] NAME[:TAG]</code>
<code>-i interactive</code>		Build an image from a Dockerfile
Stop a running container		<code>docker image build [OPTIONS] PATH</code>
<code>docker container stop [container]</code>		<code>-t tag</code>
Kill a running container		
<code>docker container kill [container]</code>		

Docker File - image recipes

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 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
```

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
```

