

Glossary

Layer - a set of read-only files to provision the system

Image - a read-only layer that is the base of your container. Might have a parent image

Container - a runnable instance of the image

Registry / Hub - central place where images live

Docker machine - a VM to run Docker containers (Linux does this natively)

Docker compose - a utility to run multiple containers as a system

Useful one-liners

Download an image

`docker pull image_name`

Start and stop the container

`docker [start|stop] container_name`

Create and start container, run command

`docker run -ti --name container_name image_name command`

Create and start container, run command, destroy container

`docker run --rm -ti image_name command`

Example filesystem and port mappings

```
docker run -it --rm -p 8080:8080 -v
/path/to/agent.jar:/agent.jar -e
JAVA_OPTS="-javaagent:/agent.jar"
tomcat:8.0.29-jre8
```

Docker cleanup commands

Kill all running containers

`docker kill $(docker ps -q)`

Delete dangling images

`docker rmi $(docker images -q -f
dangling=true)`

Remove all stopped containers

`docker rm $(docker ps -a -q)`

Docker machine commands

Use docker-machine to run the containers

Start a machine

`docker-machine start machine_name`

Configure docker to use a specific machine

`eval "$(docker-machine env machine_name)"`

Docker compose syntax

docker-compose.yml file example

version: "2"

services:

web:

container_name: "web"

image: java:8 # image name

command to run

command: java -jar /app/app.jar

ports: # map ports to the host

- "4567:4567"

volumes: # map filesystem to the host

- ./myapp.jar:/app/app.jar

mongo: # container name

image: mongo # image name

Create and start containers

`docker-compose up`

Interacting with a container

Run a command in the container

`docker exec -ti container_name command.sh`

Follow the container logs

`docker logs -ft container_name`

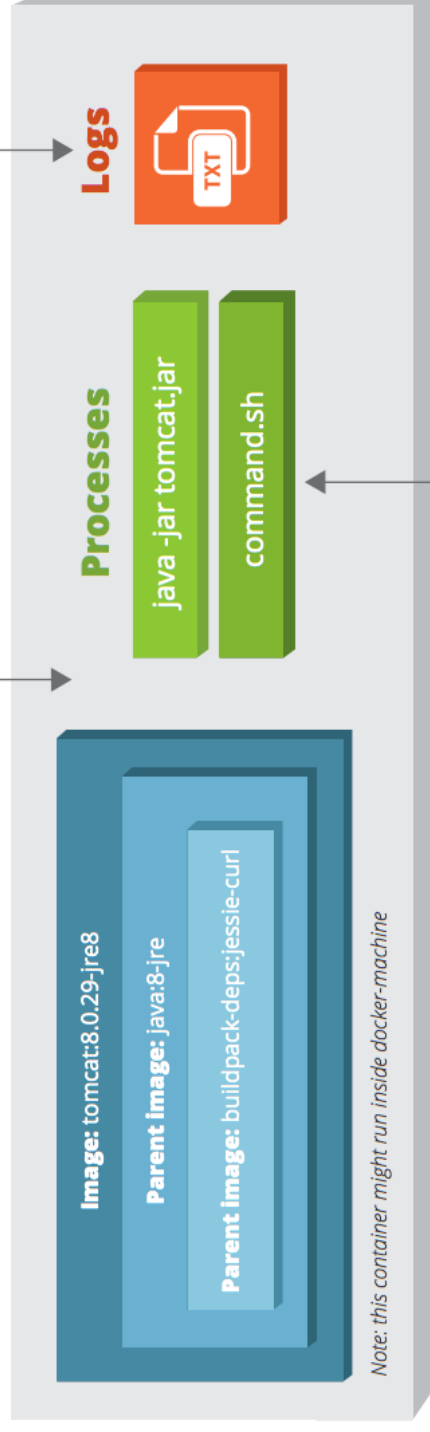
Save a running container as an image

`docker commit -m "commit message" -a "author"
container_name username/image_name:tag`

Container: my-container

`docker start my-container`

`docker logs -ft my-container`



`docker exec -ti my-container command.sh`