

ThoughtWorks®

DOCKER

Rise of the Containers Workshop

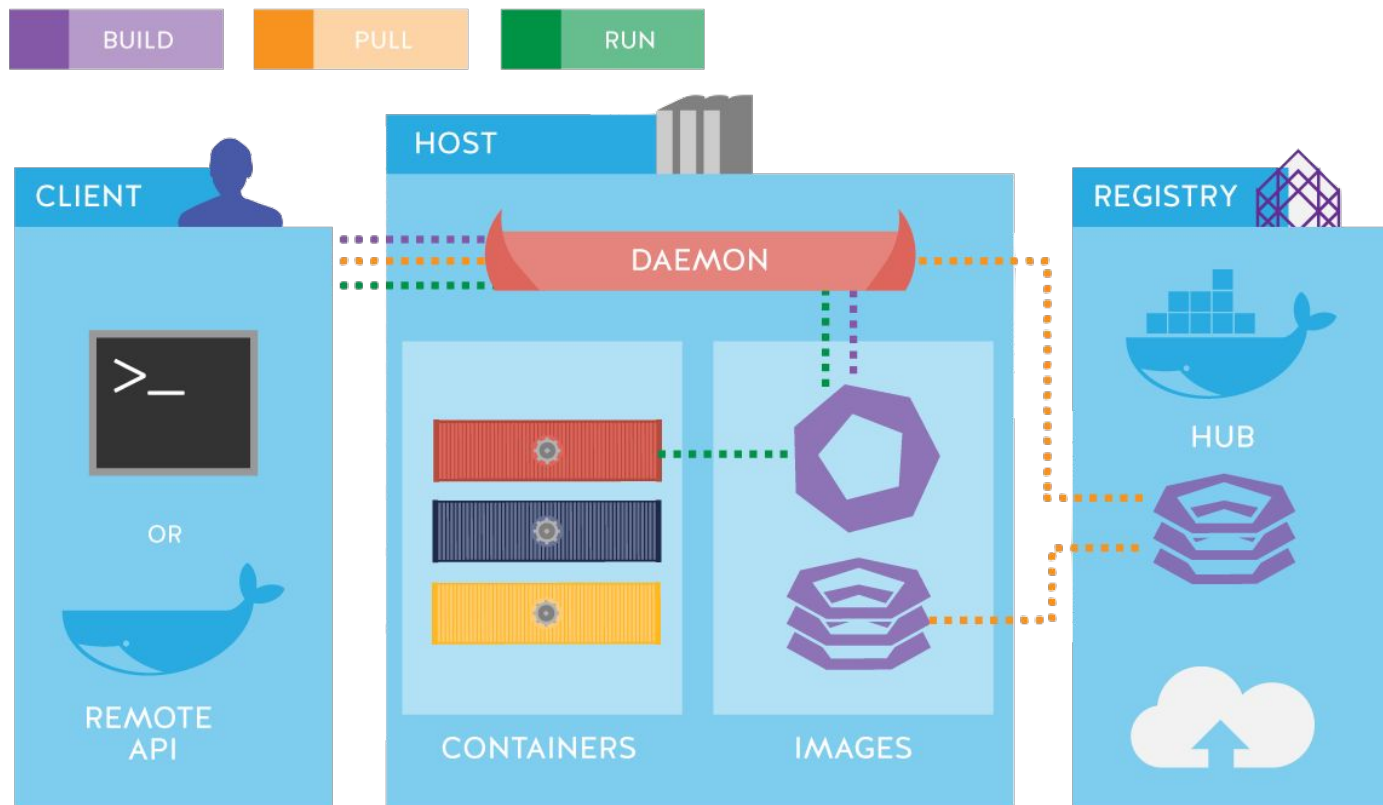
LXC

Tightly Coupled with Host
Persistent
Multi App env
No Versioning / Reusability

Docker

Portable
Ephemeral
Single Application
Reusable versioned Layers

Docker architecture



Basic Demo

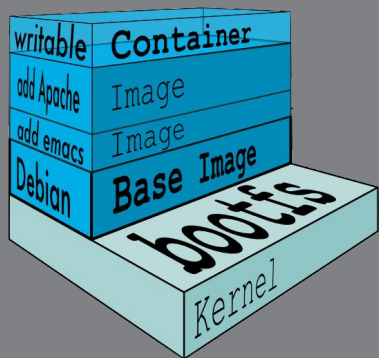
docker images

docker ps

docker pull

docker run

docker [start/stop]



Docker Images

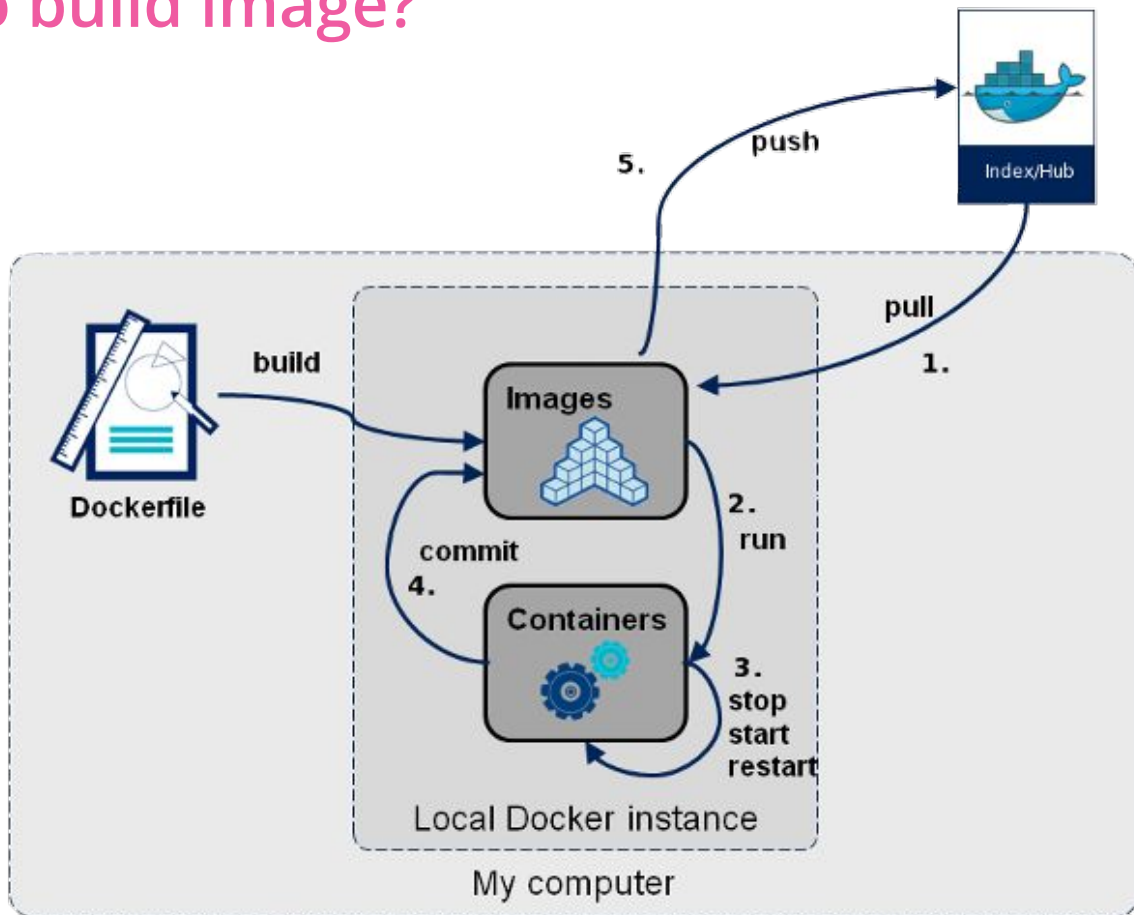
An image is an inert, immutable, file that's essentially a snapshot of a container. Images are stored in a Docker registry such as registry.hub.docker.com.

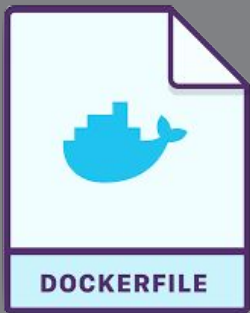
Docker image layers

Runtime file system (read-write)
WAR file added (read-only)
Tomcat installed (read-only)
Java 8 installed (read-only)
Apache HTTP installed (read-only)
Debian Wheezy (read-only base image)
Linux Kernel 3.13.3 (read-only bootfs)

ID: ca1f5f48ef43 Parent: 91bac885982d Name: my_image:1.0
ID: 91bac885982d Parent: 3df5aff384fc Name: ""
ID: 3df5aff384fc Parent: a719479f5894 Name: ""
ID: a719479f5894 Parent: "" Name: ""

How to build image?





Dockerfile

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Dockerfile

```
FROM openjdk:alpine // image to build from

// copy file(s) from host to image
[ADD/COPY] /src/path/from/host /dest/path/to/image

// executes command(s) as new layer, used to install softwares
RUN apt-get install python3 \
    git

EXPOSE 8080 9009 // expose ports from container

// configures a container that runs as an executable, main process start
ENTRYPOINT ["executable", "param1", "param2", ...]
```

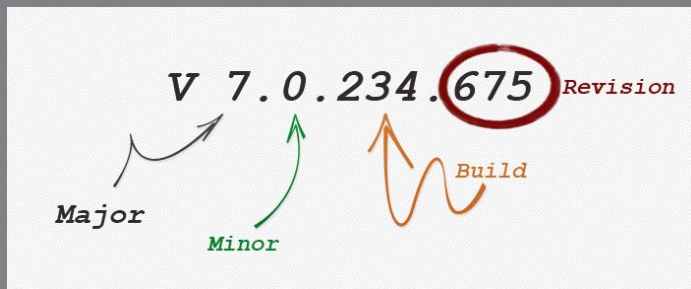
Dockerfile Demo

docker images

docker build

docker rmi

incl. image layers



Versioning Docker Images

As you're iterating on your application, you'll need to push new Docker images to the registry. A natural question that comes is how to version these images?

Docker image name & version

```
~/Projects/boot-services/metadata-service ➤ master docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
metadata-service	1.0.0.7638113	6a5e8db6f6bb	25 seconds ago	126MB
metadata-service	latest	6a5e8db6f6bb	25 seconds ago	126MB
k8s.gcr.io/kube-proxy-amd64	v1.10.0	bfc21aadc7d3	5 weeks ago	97MB
k8s.gcr.io/kube-scheduler-amd64	v1.10.0	704ba848e69a	5 weeks ago	50.4MB
k8s.gcr.io/kube-controller-manager-amd64	v1.10.0	ad86dbed1555	5 weeks ago	148MB
k8s.gcr.io/kube-apiserver-amd64	v1.10.0	af20925d51a3	5 weeks ago	225MB
k8s.gcr.io/etcd-amd64	3.1.12	52920ad46f5b	7 weeks ago	193MB
k8s.gcr.io/kube-addon-manager	v8.6	9c16409588eb	2 months ago	78.4MB
openjdk	alpine	224765a6bdbe	3 months ago	102MB

k8s.gcr.io/kube-proxy-amd64:v1.10.0

namespace

image Name (application name)

tag (version)

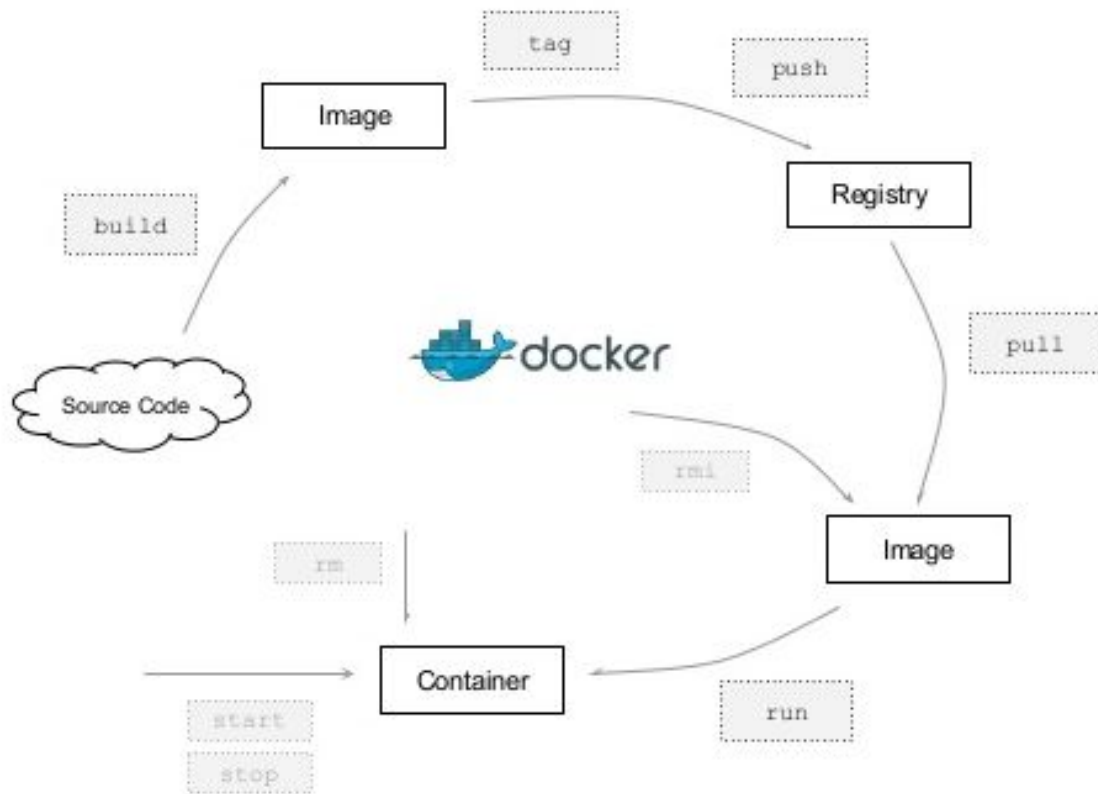
Image tagging Demo

docker tag

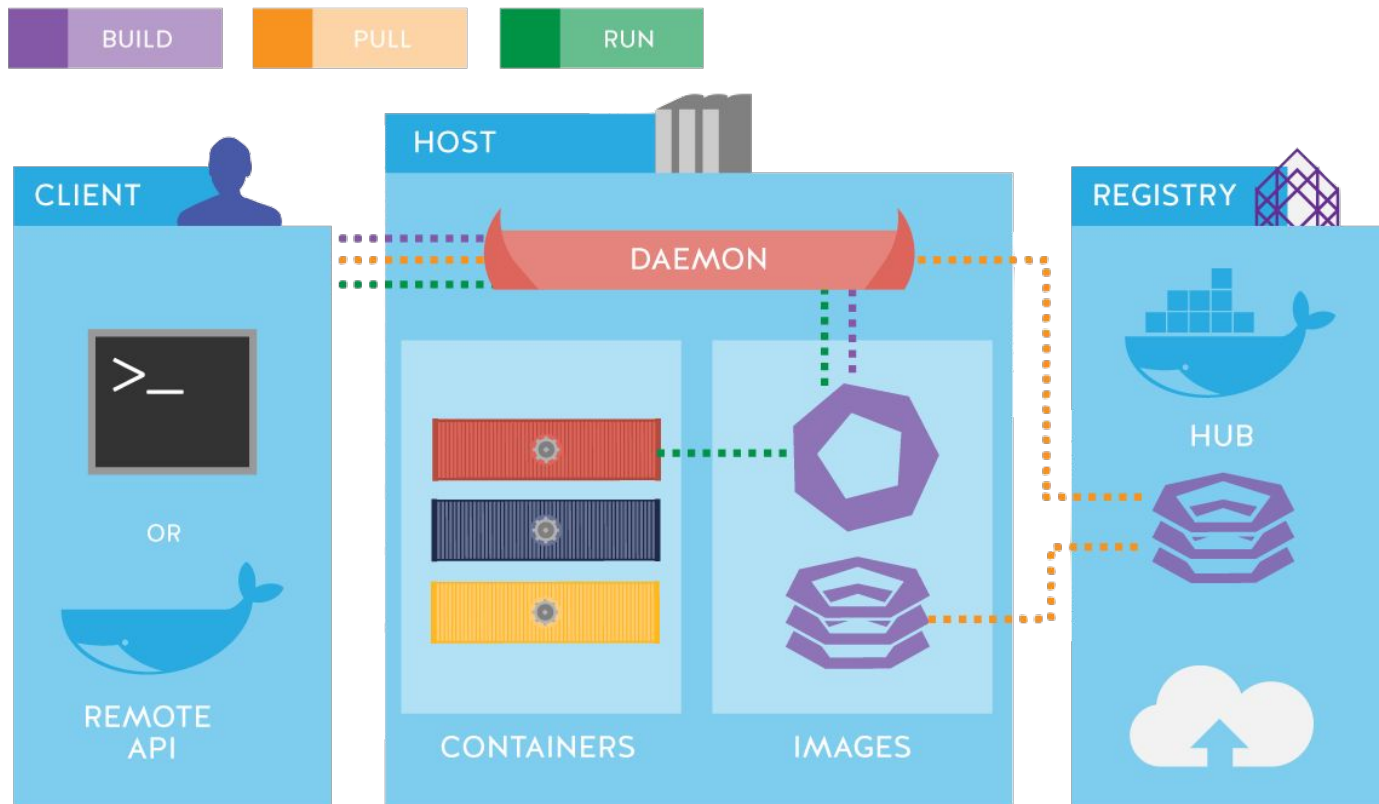
docker login

docker push

Docker workflow



Revisiting docker architecture





Docker for Developers

THANK YOU

For questions or suggestions:

Kulvinder Singh

kulvinds@thoughtworks.com