



A Gentle Introduction to Docker_

Marko Radinovic

06.07.2016

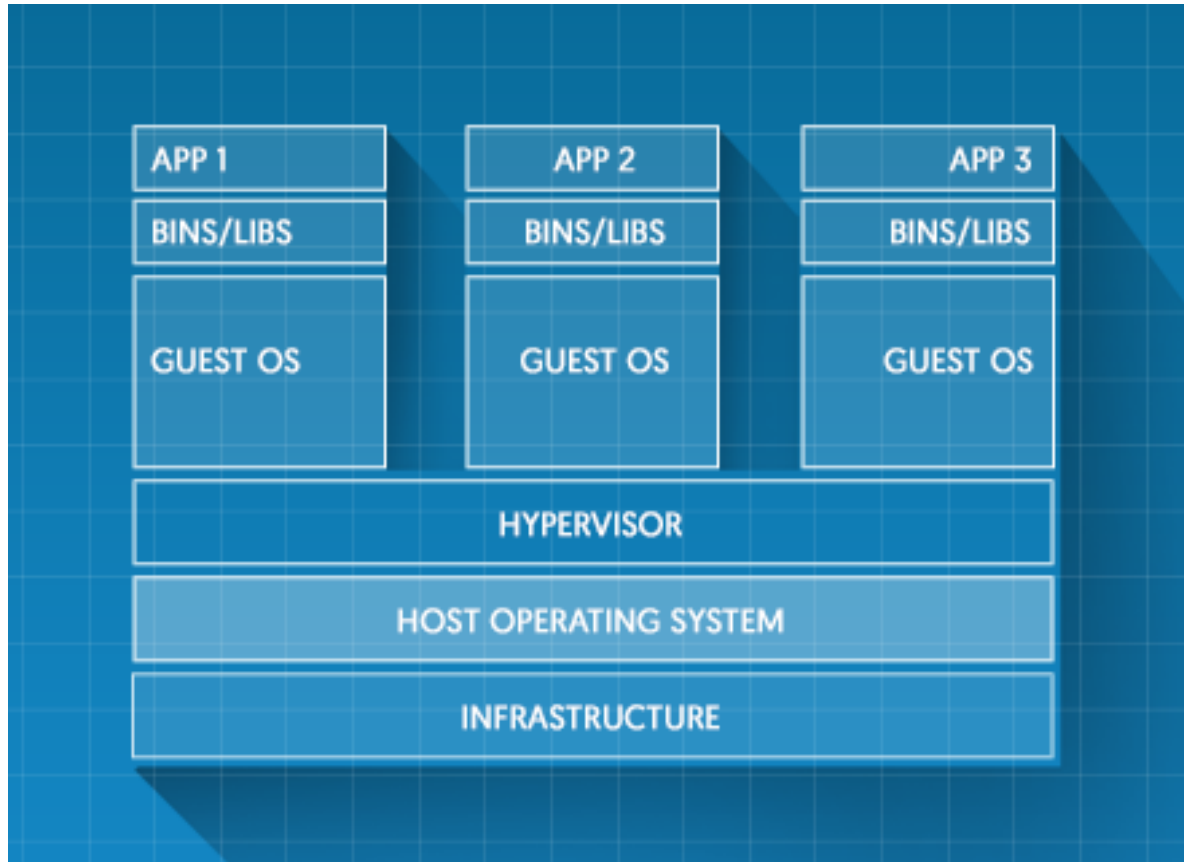
What is Docker?

PACKAGE YOUR APPLICATION INTO A STANDARDIZED UNIT FOR SOFTWARE DEVELOPMENT

Docker is an open platform for developing, shipping, and running applications.

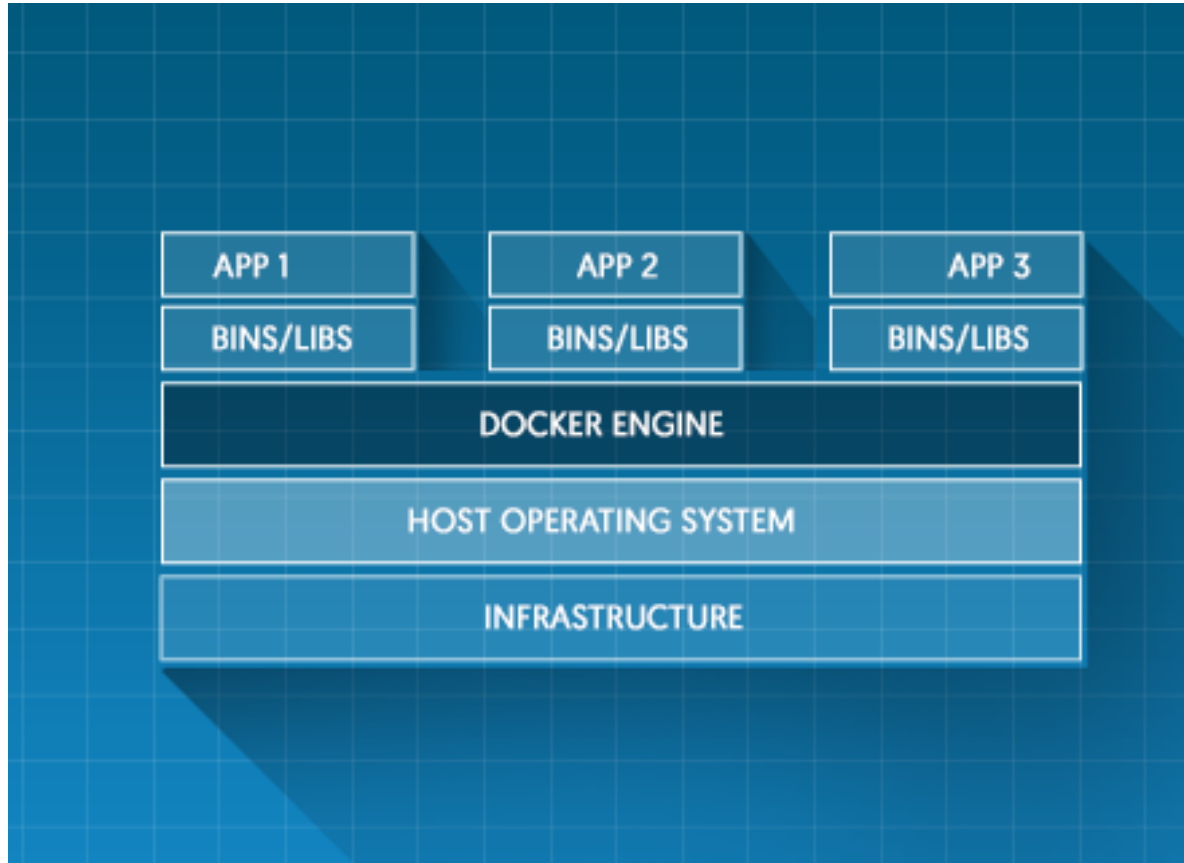
BUILD – SHIP – RUN

Virtual Machines



Virtual machines include the application, the necessary binaries and libraries, and an entire guest operating system.

Containers



Containers include the application and all of its dependencies-but share the kernel with other containers, running as isolated processes on the host operating system.

Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server.

Lightweight

They start instantly and use less RAM. Images are constructed from layered filesystems and share common files, making disk usage and image downloads much more efficient.

Open

Docker containers are based on open standards, enabling containers to run on all major operating systems and on top of any infrastructure.

Secure by default

Containers isolate applications from one another and the underlying infrastructure, while providing an added layer of protection for the application.

What can I use Docker for?

- Faster delivery of your application (CI)
- Deploying and scaling more easily (CD)
- Achieving higher density and running more workloads

Docker Engine

What is Docker Engine?

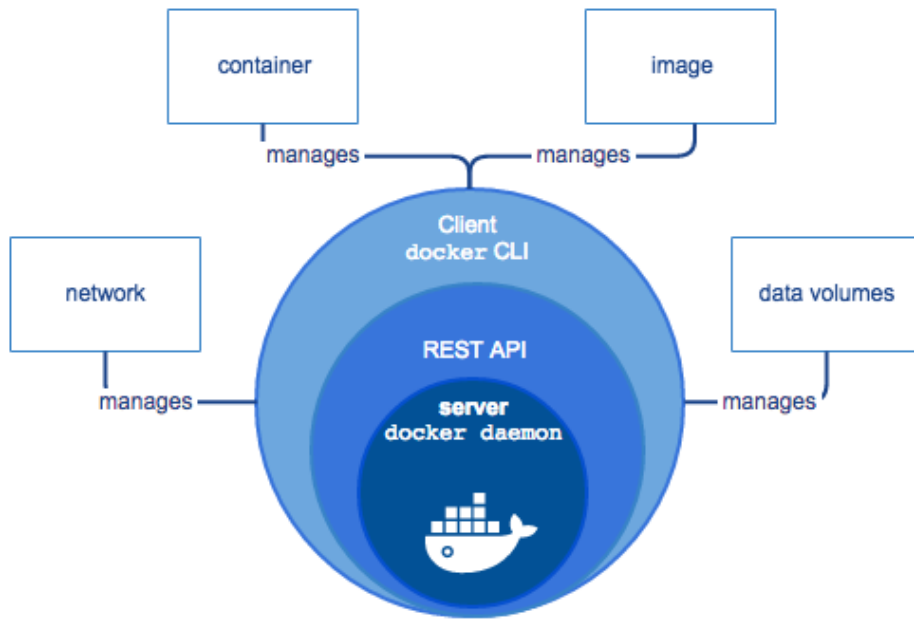
Docker Engine is a client-server application with these major components:

A server which is a type of long-running program called a daemon process.

A REST API that programs can use to talk to the daemon and instruct it what to do.

A command line interface (CLI) client.

Docker Engine



The CLI makes use of the Docker REST API to control or interact with the Docker daemon.

Many other Docker applications make use of the underlying API and CLI.

The daemon creates and manages Docker objects.

Docker objects include images, containers, networks, data volumes, and so forth.

A Docker image is a read-only template.
Images are used to create Docker containers.

```
Container c = new Image("ubuntu");
```

Docker images are the build component of
Docker.

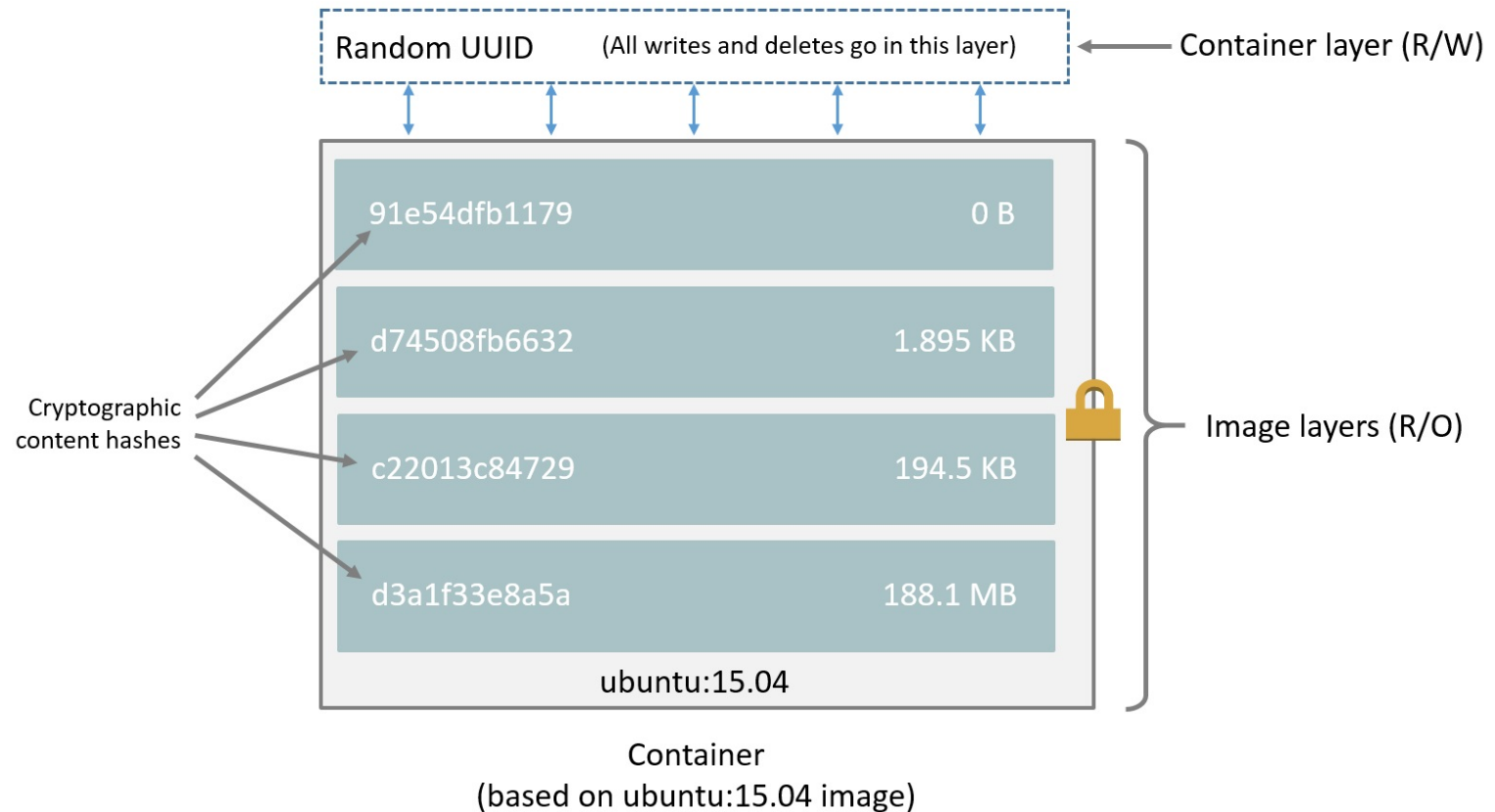
Let's run some containers

Docker images are **read-only** templates from which Docker containers are launched.

Each image consists of a series of layers.

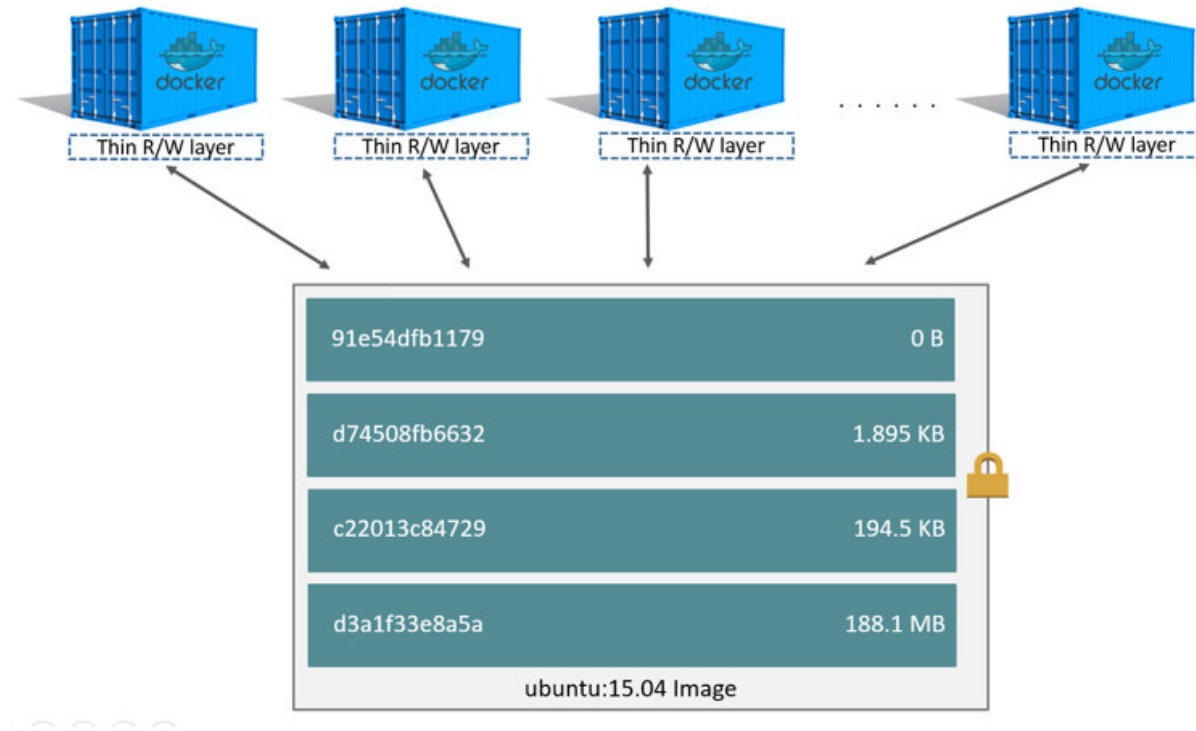
Docker makes use of union file systems to combine these layers into a single image.

Docker Images



Each Docker image references a list of read-only layers that represent filesystem differences. Layers are stacked on top of each other to form a base for a container's root filesystem.

Docker Images



Because each container has its own thin writable container layer, and all changes are stored in this container layer, this means that multiple containers can share access to the same underlying image and yet have their own data state.

How to build Docker images?

Update and commit an image (don't do this)



Build image from Dockerfile

Dockerfile

Dockerfile contains a set of instructions that tell Docker how to build our image.

```
FROM ubuntu:latest
MAINTAINER Marko Radinovic
RUN apt-get update
RUN apt-get install -y nginx
COPY index.html /usr/share/nginx/html/
ENTRYPOINT ["/usr/sbin/nginx","-g","daemon off;"]
EXPOSE 80
```

General guidelines and recommendations

- Containers should be ephemeral
- Use a .dockerignore file
- Avoid installing unnecessary packages
- **Run only one process per container**
- Minimize the number of layers
- Sort multi-line arguments

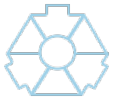
Let's build some images

Docker Universe



Docker Engine

Create Docker images and run Docker containers.



Docker Hub

A hosted registry service for managing and building images.



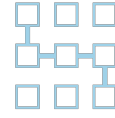
Docker Trusted Registry

(DTR) stores and signs your images.



Docker Machine

Automate container provisioning on your network or in the cloud. Available for Windows, Mac OS X, or Linux..



Docker Compose

Defines applications built using multiple containers.



Docker Cloud

A hosted service for building, testing, and deploying Docker images to your hosts.



Docker Universal Control Plane

(UCP) Manage a cluster of on-premises Docker hosts as if they were a single machines.

Docker Compose

Docker Compose

Compose is a tool for defining and running multi-container Docker applications.

With Compose, you use a Compose file to configure your application's services.

docker-compose.yml

```
version: '2'
services:
  web:
    image: web
    depends_on:
      - db
  db:
    image: postgres
```

Q&A