



docker

Workshop L1



Our Host:

OFFICE **12**
THE COWORKING ONE

<https://www.facebook.com/office12.gr>

<http://bit.ly/office12gmaps>

WHAT IS DOCKER

Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud.

Docker is a containerization platform which packages your application and all its dependencies together in the form of containers so as to ensure that your application works seamlessly in any environment be it development or test or production.

BACK IN TIME...

1979: unix V7 -> chroot

2000: FreeBSD -> Jails

2001: Linux Vserver

2004: Oracle Solaris Containers (ZFS) (kernel patches)

2005: Open VZ (Open Virtuozzo) (kernel patches – non public)

2006: Process Containers (GOOGLE control groups === cgroups)

2007: cgroups merged into 2.6.24 kernel !!!

2008: LXC (first most complete container manager)

BACK IN TIME...

2011: Warden (cloudfoundry)

2013: LMCTFY (2015 stopped -> GOOGLE to OCF)

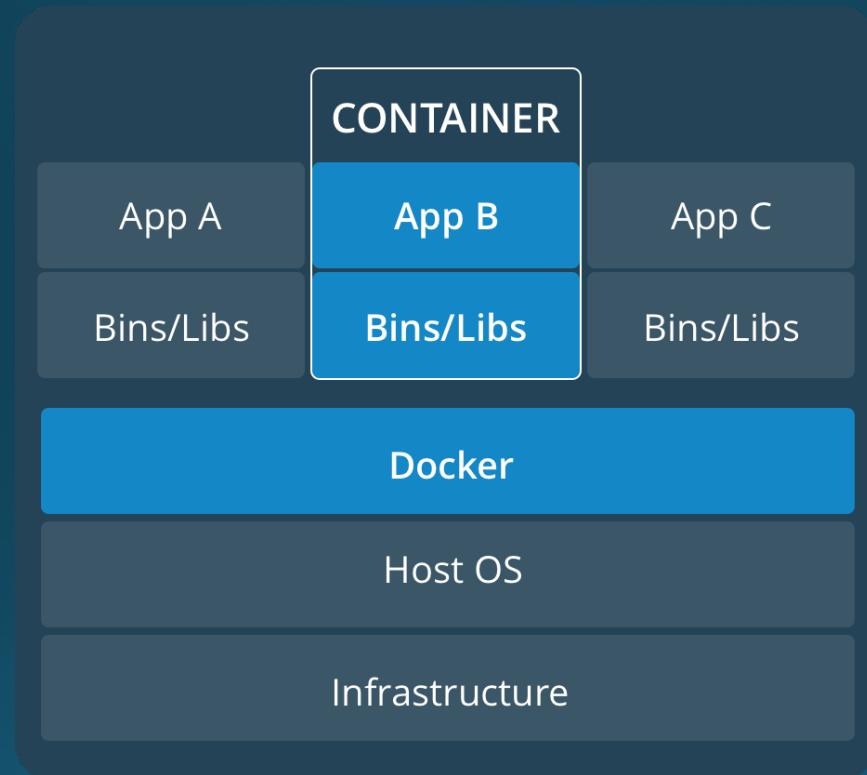
2013: Docker

2016: ConSec

2017: Containers become mature, rkt, Containerd, Kubernetes

WHAT IS A DOCKER CONTAINER

Docker containers, wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries etc. anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.



WHAT IS A DOCKER IMAGE

Docker image is the source of Docker container. In other words, Docker images are used to create containers. Images are created with the build command, and they'll produce a container when started with run.

Images are stored in a Docker registry such as registry.hub.docker.com because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

WHAT IS DOCKER HUB

Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts.

It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

WHAT IS A DOCKER SWARM

Docker Swarm can be best understood as the native way of Clustering implementation for Docker itself. Docker Swarm turns a pool of Docker hosts into a single and virtual Docker host. It serves the standard Docker API or any other tool that can already communicate with a Docker daemon can make use of Docker Swarm to scale in a transparent way to multiple hosts.

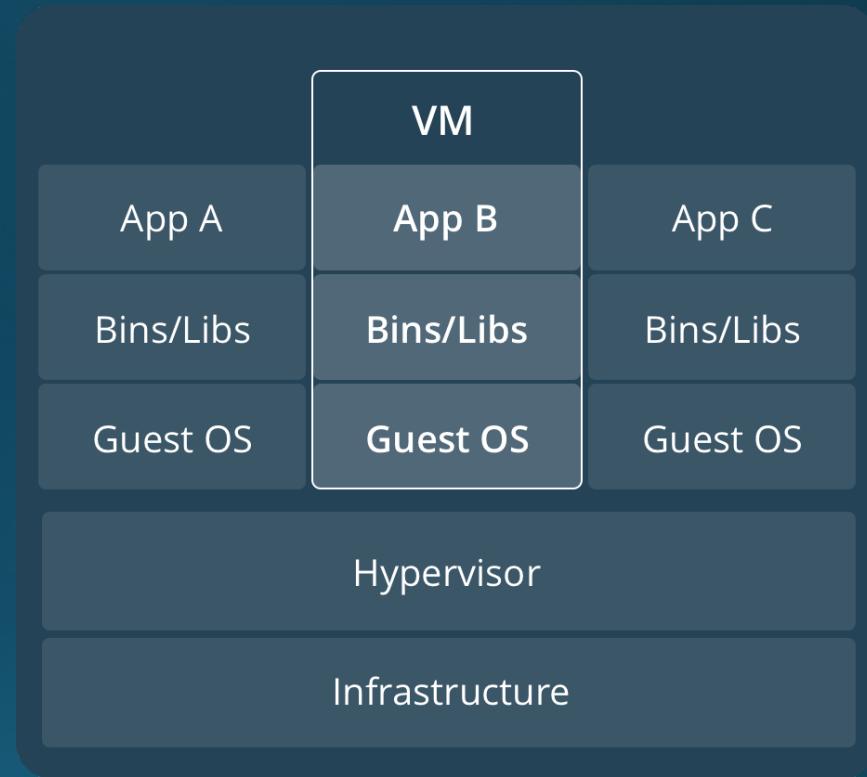
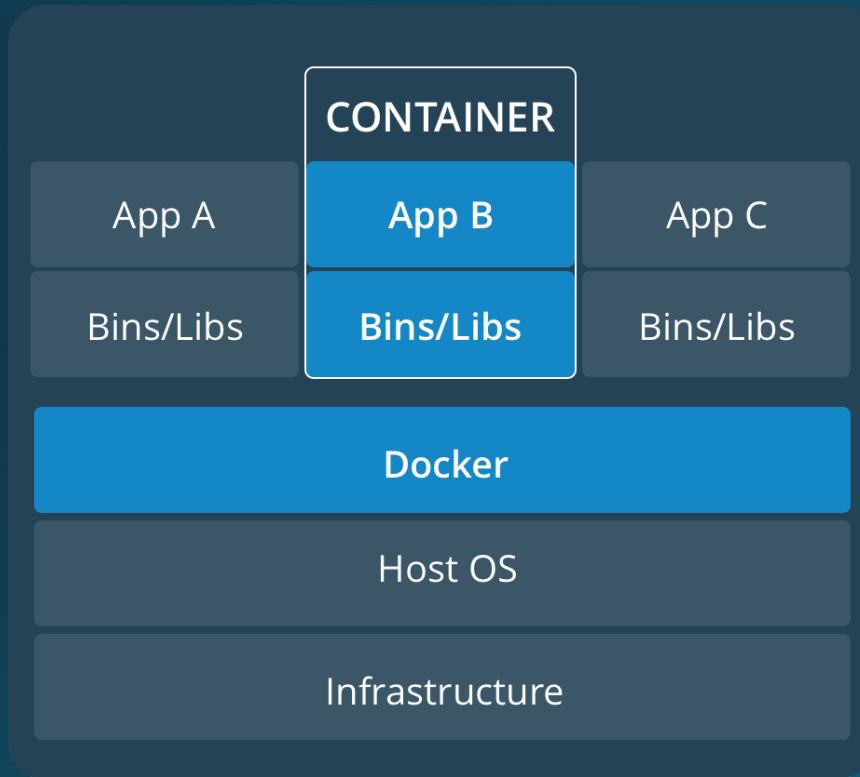
WHAT IS A CONTAINER PLATFORM

A container platform is a complete solution that allows organizations to solve multiple problems across a diverse set of requirements. It is more than a piece of technology and orchestration - it delivers sustainable benefits throughout your organization by providing all the pieces an enterprise operation requires including security, governance, automation, support and certification over the entire application lifecycle.

CONTAINER VS VIRTUAL MACHINE

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware.

Containers are more portable and efficient.



Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), and start almost instantly.

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries - taking up tens of GBs. VMs can also be slow to boot.

WHAT IS A HYPERVISOR

A hypervisor is a process that separates a computer's operating system and applications from the underlying physical hardware. Usually done as software although embedded hypervisors can be created for things like mobile devices.

The hypervisor drives the concept of virtualization by allowing the physical host machine to operate multiple virtual machines as guests to help maximize the effective use of computing resources such as memory, network bandwidth and CPU cycles.

WHAT IS A UNIKERNEL

A unikernel is an executable image that can execute natively on a hypervisor, without the need for a separate operating system.

The image contains application code, as well as all the operating system functions required by that application.

WHAT IS A KATA CONTAINER

Kata Containers is an open source project and community working to build a standard implementation of lightweight Virtual Machines (VMs) that feel and perform like containers, but provide the workload isolation and security advantages of VMs.

<https://katacontainers.io/>

DOCKER CONTAINERS

WHAT IS A DOCKERFILE

Dockerfile is nothing but a set of instructions that have to be passed on to Docker itself, so that it can build images automatically reading these instructions from that specified Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Using docker build users can create an automated build that executes several command-line instructions in succession.

SAMPLE DOCKERFILE

OFFICIAL REPOSITORY FOR HTTPD

<https://github.com/docker-library/httpd/blob/3af22f20b54ccb7c76ef4c3b12bd9dcf6cae862b/2.4/Dockerfile>

<http://bit.ly/httpd-2-4-debian>

A DOCKERFILES' JOURNEY

```
# A basic apache server.  
# small image  
FROM httpd:2.4.33-alpine  
MAINTAINER Thanassis Zografos  
  
EXPOSE 80
```

A DOCKERFILES' JOURNEY

```
# A basic apache server.  
# larger image  
FROM httpd:2.4.33  
MAINTAINER Thanassis Zografos  
  
EXPOSE 80
```

LET'S BUILD

```
docker build -t sonaht:apache2-small --rm -f Dockerfile.small .
```

```
docker build -t sonaht:apache2-large --rm -f Dockerfile.large .
```

LETS SEE SIZES

```
sonaht@192 ▶ ~/Projects/Docker/workshop-l1/dockerfiles/httpd ▶ docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
sonaht              apache2-large   99e9cfa347a7    Less than a second ago  178MB
sonaht              apache2-minimal  70b9e07739cc    12 minutes ago       90.7MB
```

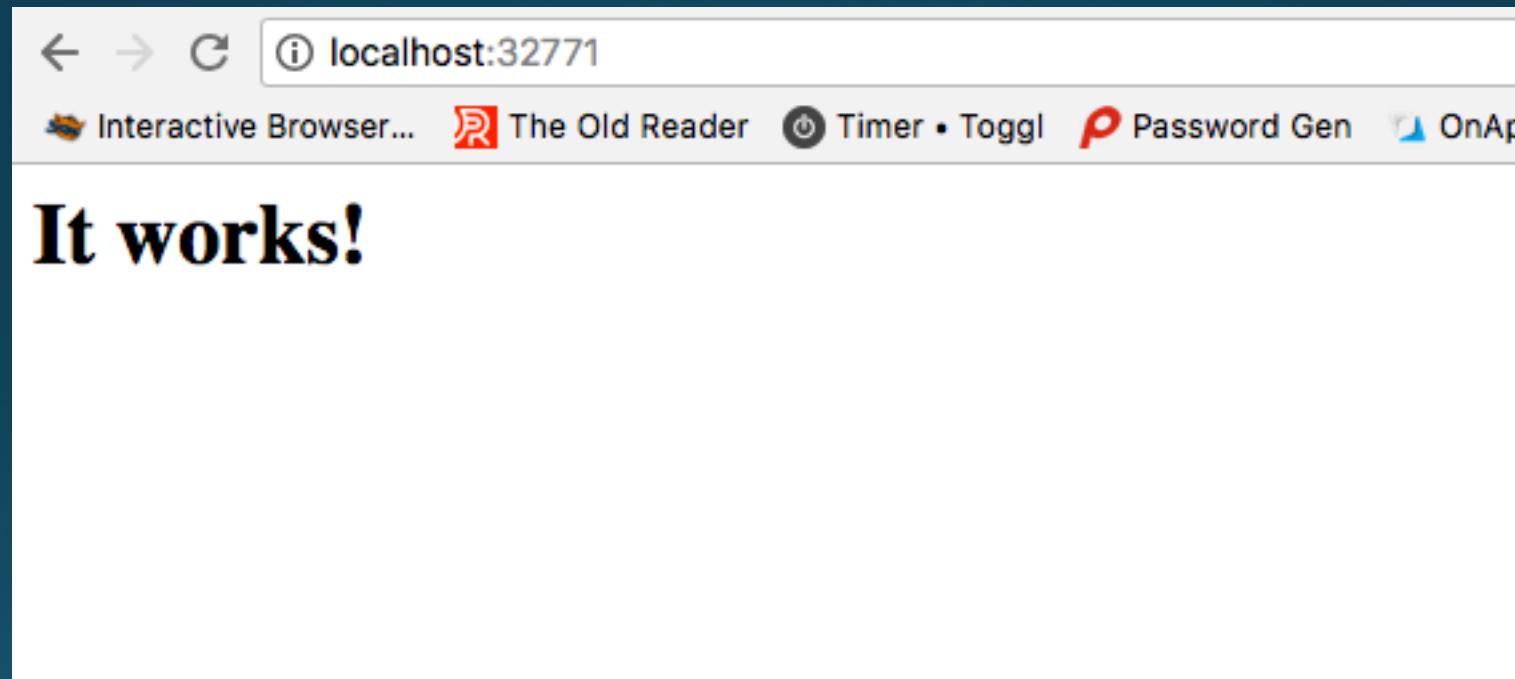
LET'S RUN

```
docker run -dit --name apache2-small -P sonaht:apache2-small
```

LET'S PS

```
sonaht@192 ▶ ~/Projects/Docker/workshop-l1/dockerfiles/httpd ▶ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS
PORTS              NAMES
a48e53e22200        sonaht:apache2-small   "httpd-foreground"   Less than a second ago   Up 1 second
0.0.0.0:32771->80/tcp      apache2-small
```

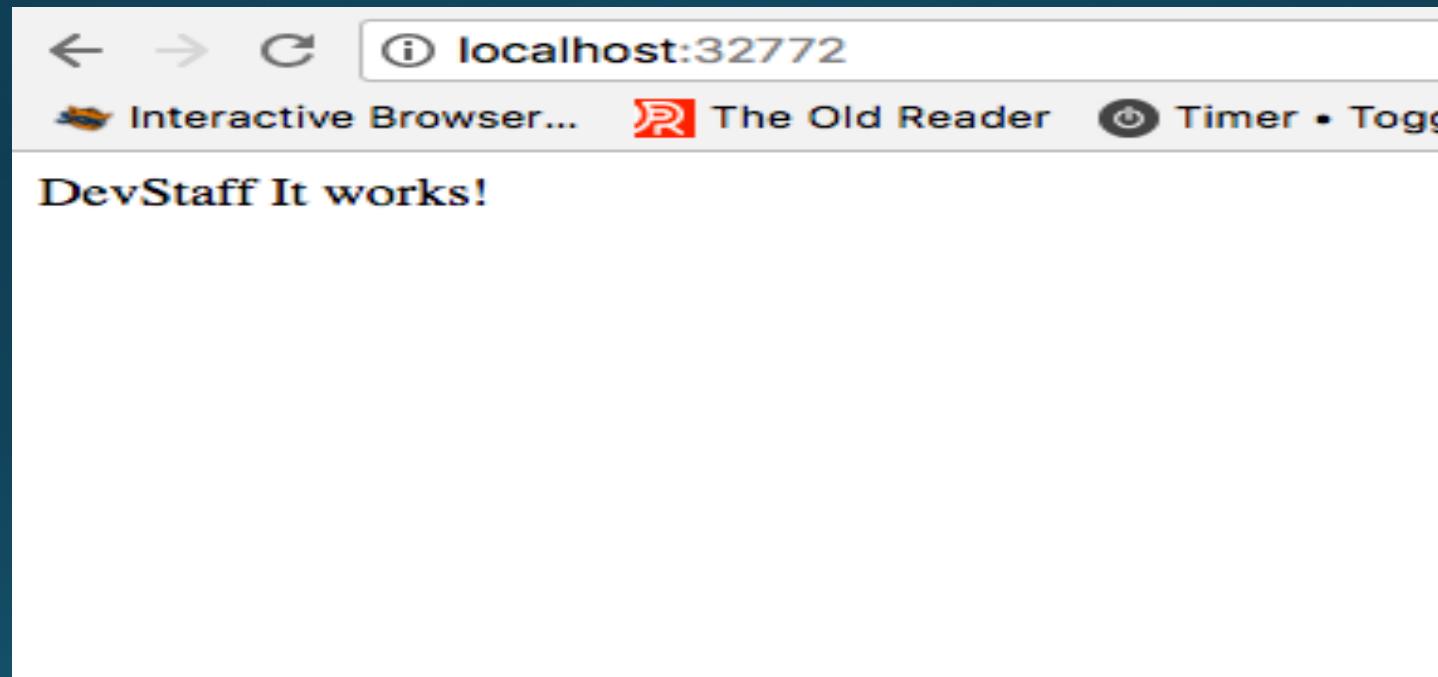
LET'S WEB



LET'S RUN DIFFERENTLY

```
docker run -dit --name apache2-small -P \  
-v "$PWD"/htdocs:/usr/local/apache2/htdocs/ sonaht:apache2-small
```

LET'S WEB DIFFERENTLY



LET'S WEB DIFFERENTLY

```
x sonaht@192 ▶ ~/Projects/Docker/workshop-l1/dockerfiles/httpd ▶ cat htdocs/index.html  
DevStaff It works!
```

LET'S RUN DIFFERENTLY 2

```
docker run -dit --name apache2-small -p8080:80 \
-v "$PWD"/htdocs:/usr/local/apache2/htdocs/ sonaht:apache2-small
```

LET'S PS DIFFERENTLY 2

```
sonaht@192 ▶ ~/Projects/Docker/workshop-l1/dockerfiles/httpd ▶ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS
PORTS
6077b7370474        sonaht:apache2-small   "httpd-foreground"   Less than a second ago   Up 1 second
0.0.0.0:8080->80/tcp      apache2-small
```

LET'S BUILD KEYWORDS 1

FROM

LABEL

RUN

APT-GET ISSUE

PIPES SPECIALITY

<https://dockr.ly/2kEXCzy>

LET'S BUILD KEYWORDS 2

CMD

EXPOSE

ENV

ADD / COPY

LET'S BUILD KEYWORDS 3

ENTRYPOINT

VOLUME

USER

WORKDIR

ONBUILD

LET'S COMPOSE A LITTLE

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

With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

LET'S COMPOSE A LITTLE EXERCISE

<https://docs.docker.com/compose/gettingstarted/>

EXERCISE 1

<https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md>

Or

<http://bit.ly/workshop-l1-ex1>

EXERCISE 2

<https://github.com/docker/labs/blob/master/beginner/chapters/websites.md>

Or

<http://bit.ly/workshop-l1-ex2>

EXERCISE 3

<https://github.com/docker/labs/blob/master/beginner/chapters/votingapp.md>

Or

<http://bit.ly/workshop-l1-ex3>