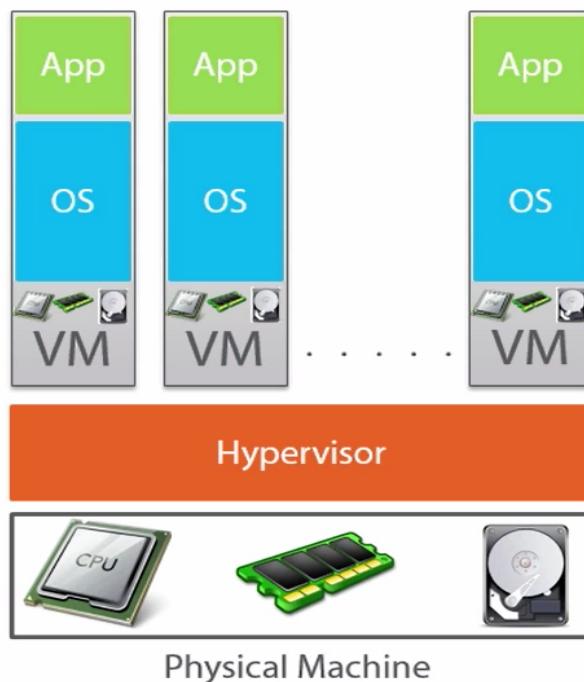


Docker

The container world

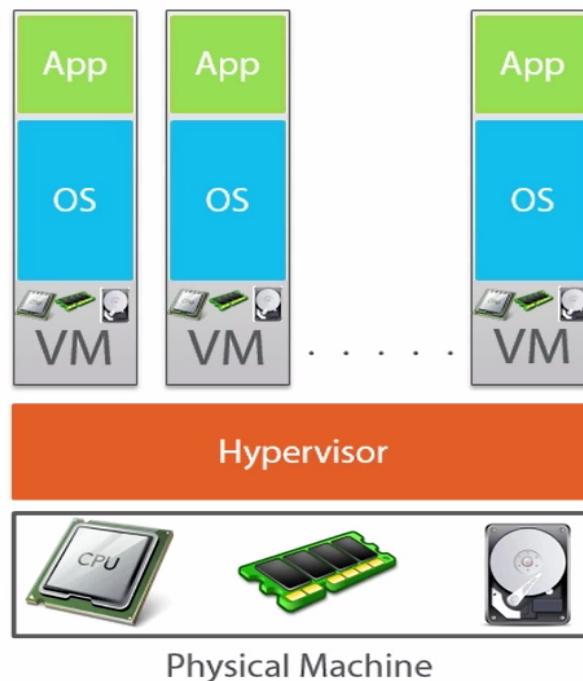
Virtual Machine vs. Container

Virtual Machine Architecture

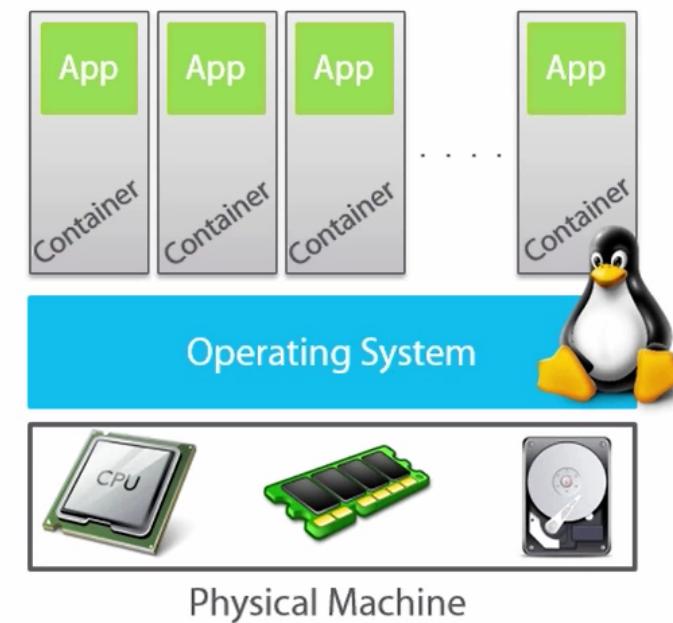


Virtual Machine vs. Container

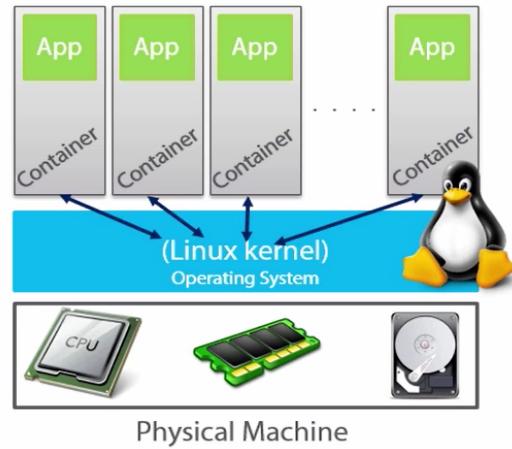
Virtual Machine Architecture



Container Architecture

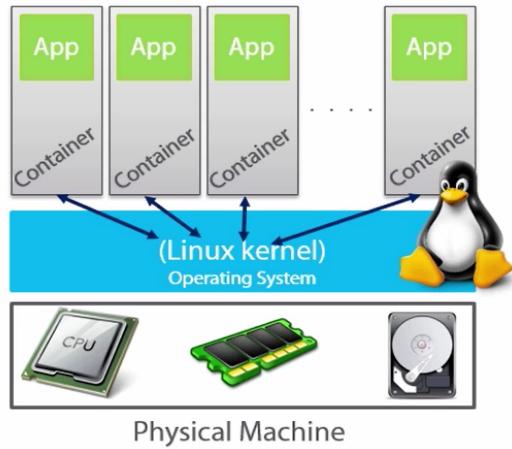


Why Containers (Docker)?

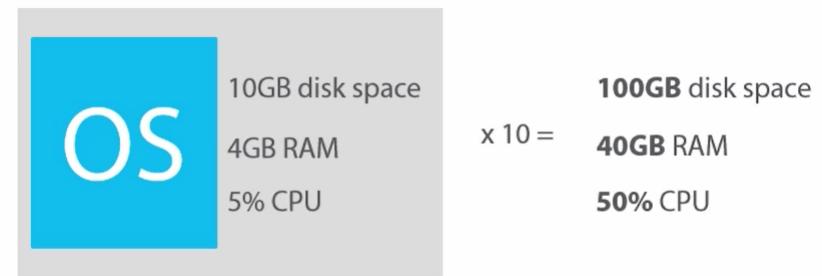


Containers consume less CPU, RAM and disk resource than Virtual Machines

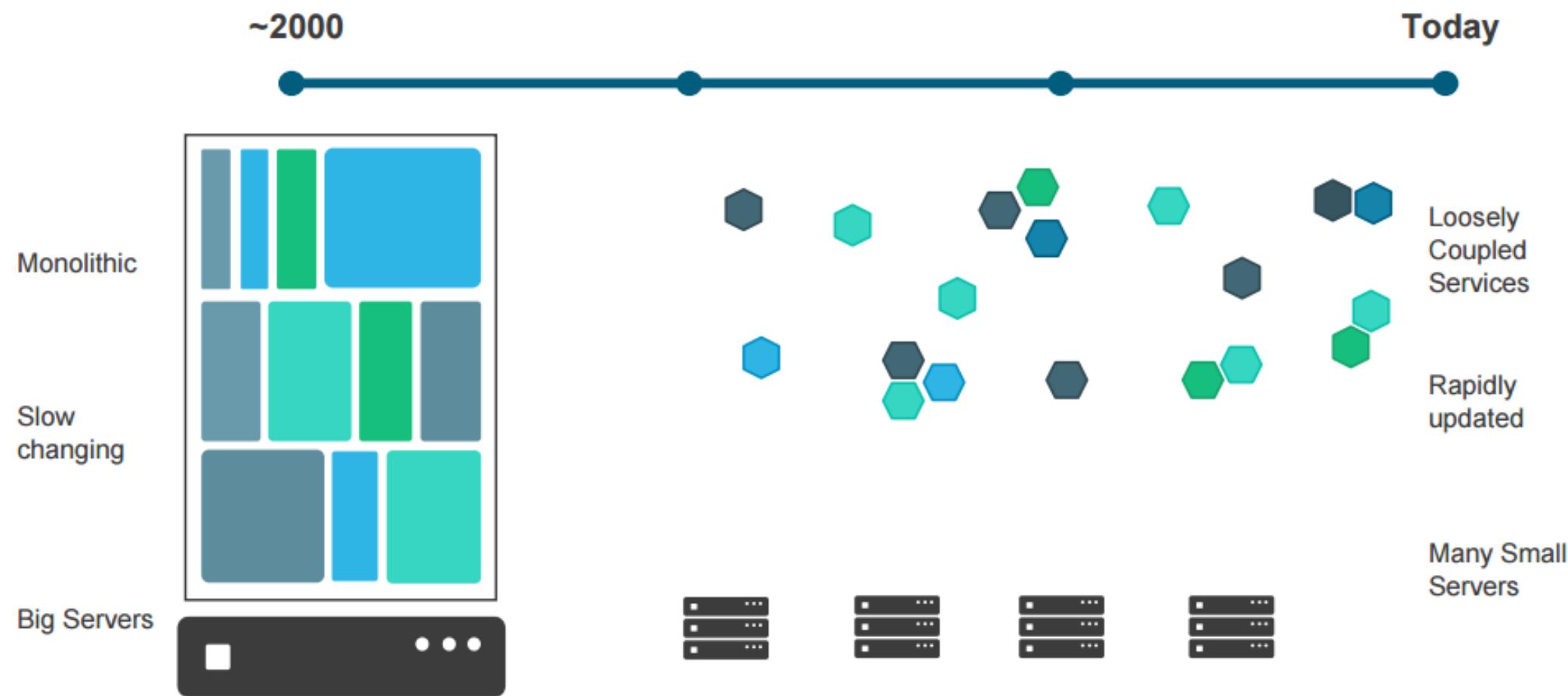
Why Containers (Docker)?



Containers consume less CPU, RAM and disk resource than Virtual Machines



A Paradigm shift in Software Development

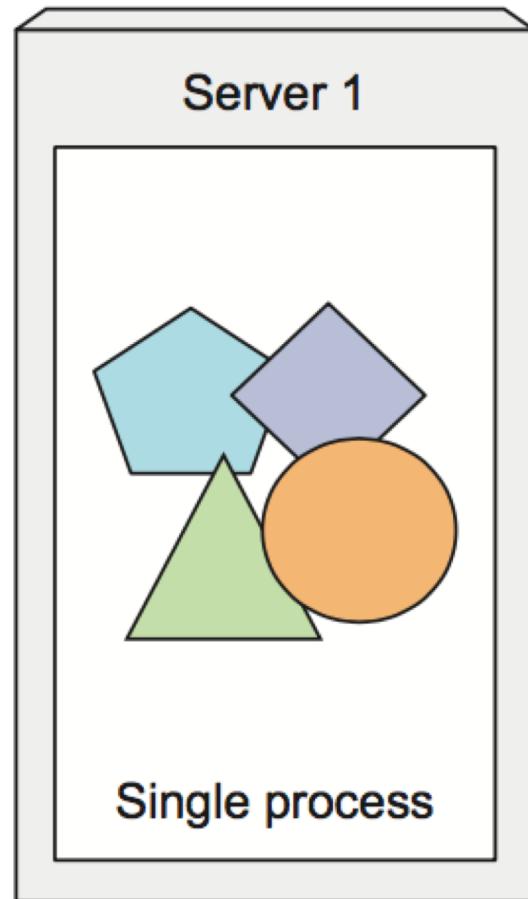


Monolith vs

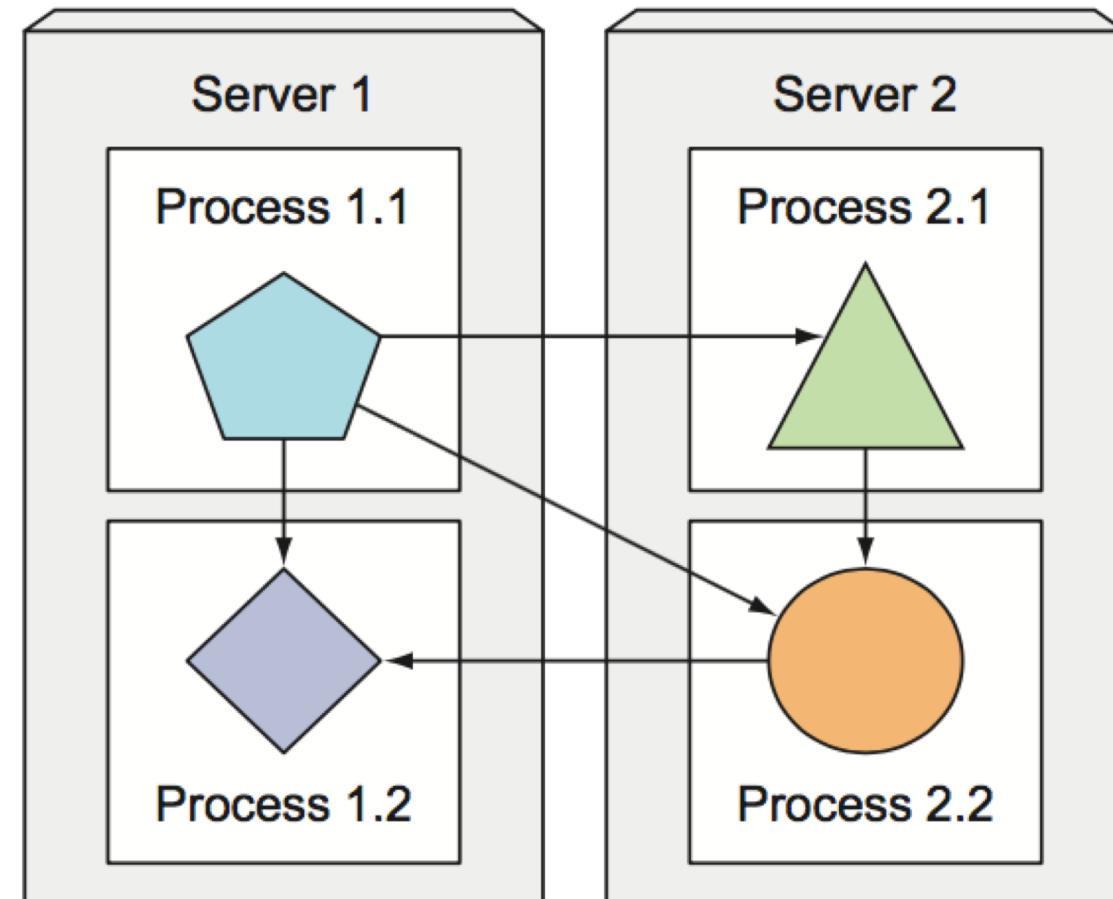
Microservices



Monolithic application



Microservices-based application



Monolith Architecture vs. Microservice Architecture

	Monolith Architecture	Microservice Architecture
Architecture	A monolith architecture represents a single logical program unit. All functions, libraries and dependencies are located within one “application block”.	A microservice architecture consists of a series of small services that work independently and communicate with each other. Each service can be used in more than one application.
Scalability	The entire application scales horizontal behind a load balancer.	Each service scales independently and on demand.
Agility	Changes to the system lead to the compilation, testing and deployment of the entire application.	Each service can be changed independently.
Development	The development based on a single programming language.	Each service can be developed in a different programming language. The integration happens via a defined API.
Maintenance	Very long and confusing source code.	Many pieces of source code that are easier to administrate.

Source: Crisp Research AG, 2015



What is Docker?

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system-level virtualization on Linux.

Docker is a client Server application where both the daemon and client can run on the same system or you can connect a Docker client with a remote Docker daemon.

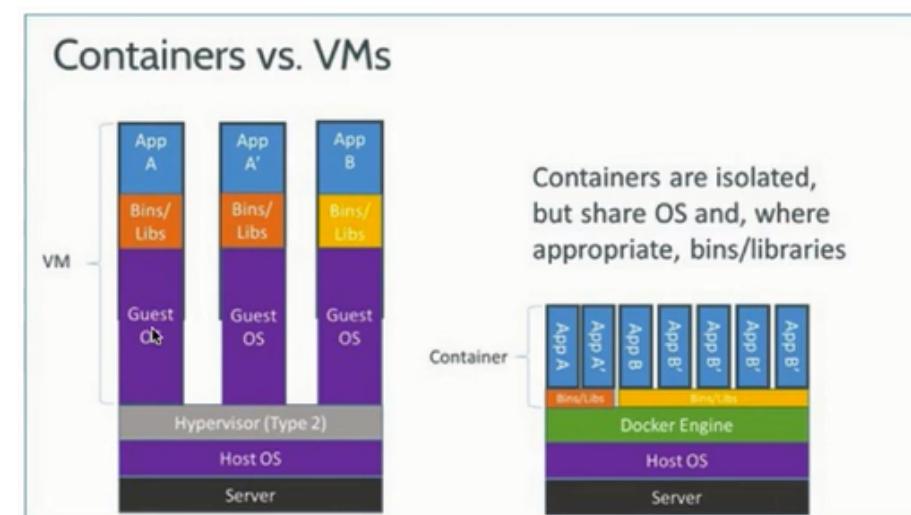
Build, Ship and Run Any App, Anywhere

Lightweight Virtualized Environments

What is Docker?....continued

Instead of virtualizing hardware, containers rest on top of a single Linux instance. This allows Docker (or generic LXC) to leave behind a lot of bloat associated with a full hardware hypervisor.

Don't mistake the Docker Engine (or the LXC process) as the equivalent of a hypervisor in a more traditional VM, it is simply the encapsulating process on the underlying system.



Hasn't this already been done?

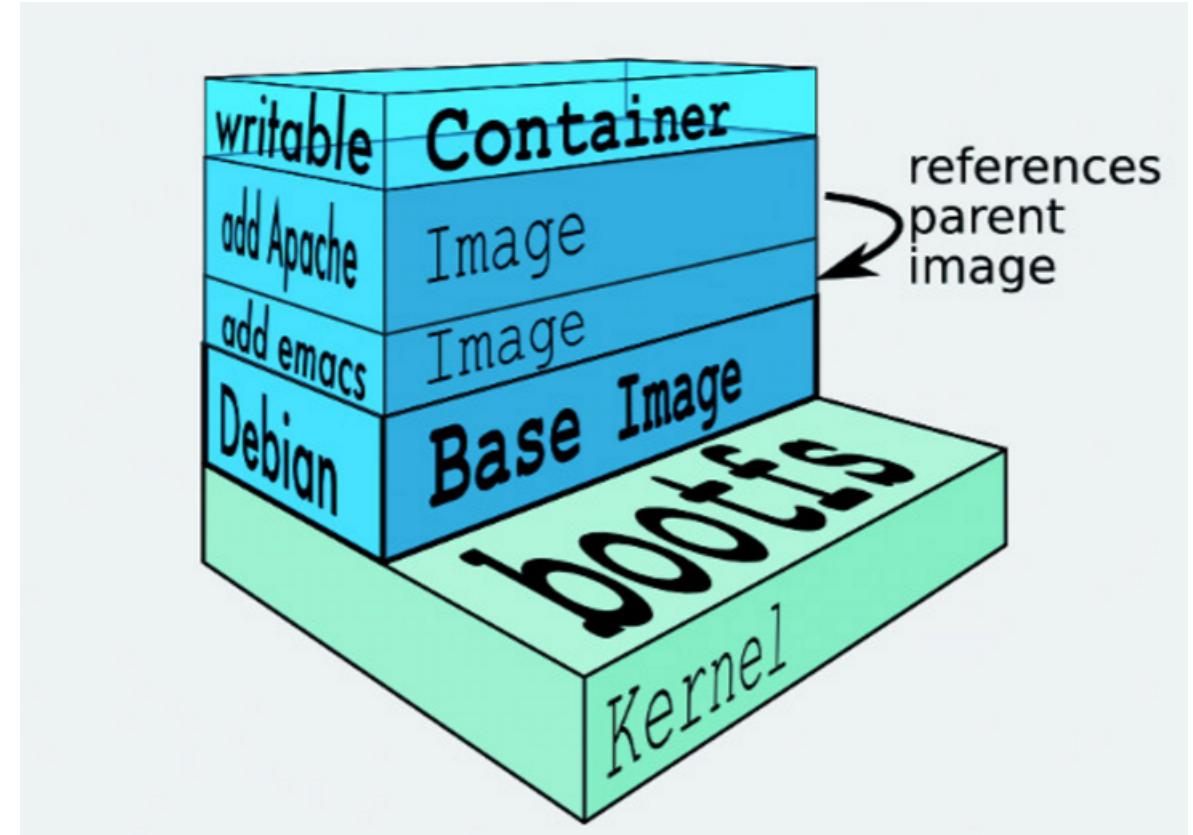
Yes, of course! Containers are not a new concept in technology, it just appears that Docker has captured the buzz (right place, right time). If you look around, you will find that a number of companies and projects have been working on the concept of application virtualization for some time:

- *FreeBSD – Jails*
- *Sun (now Oracle) Solaris – Zones*
- *Google – lmctfy (let me contain that for you)*
- *OpenVZ*

The architecture of Docker and the containers that it relies on are not new concepts, having been around since the early part of this century. However, hardware virtualization performance has now become almost indistinguishable from bare metal so that further virtualization on the technology stack can be realized.

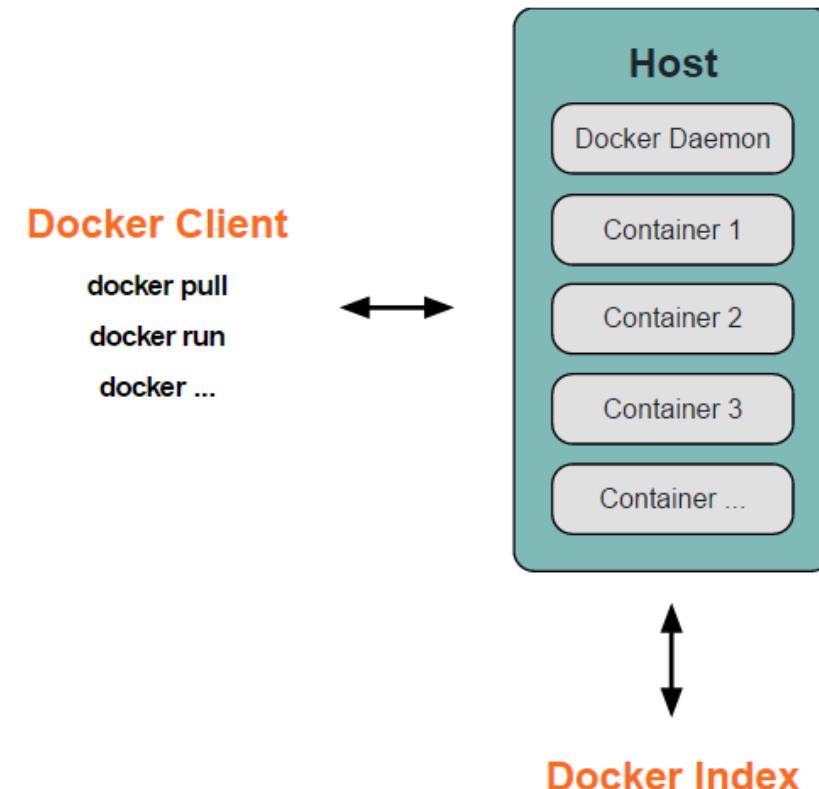
Docker Components

- [The Docker Engine](#)
- [Docker Images](#)
- [Docker Containers](#)
- [Docker Hub \(<http://hub.docker.com>\)](http://hub.docker.com)



Docker Engine

- Core of Docker: Store for containers
- Also called “docker daemon” or “runtime”
- Manages containers using jobs (similar to Unix jobs)
- Needs to be installed on all docker hosts



Docker Images

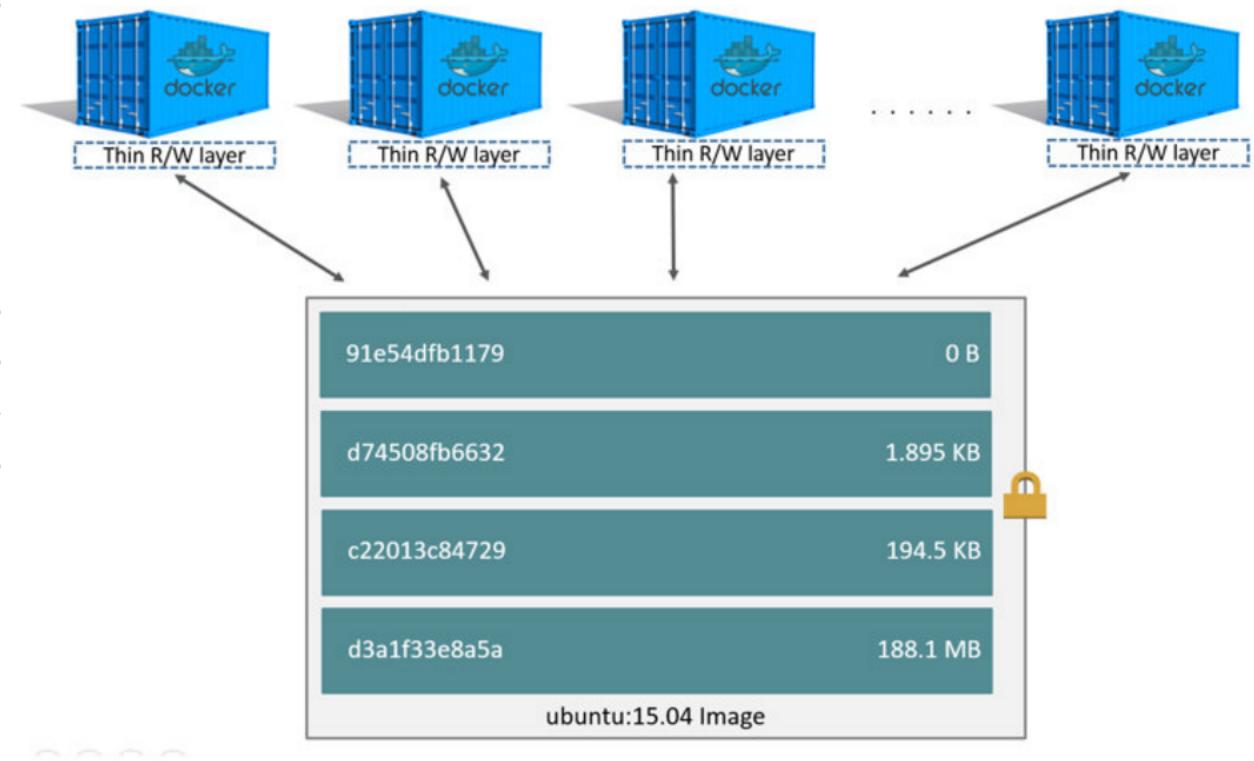
- Docker images are like templates in VM world
- Images comprise of multiple layers
- Imagename:version (defaults to "latest" if no version is mentioned)

91e54dfb1179	0 B
d74508fb6632	1.895 KB
c22013c84729	194.5 KB
d3a1f33e8a5a	188.1 MB
ubuntu:15.04	

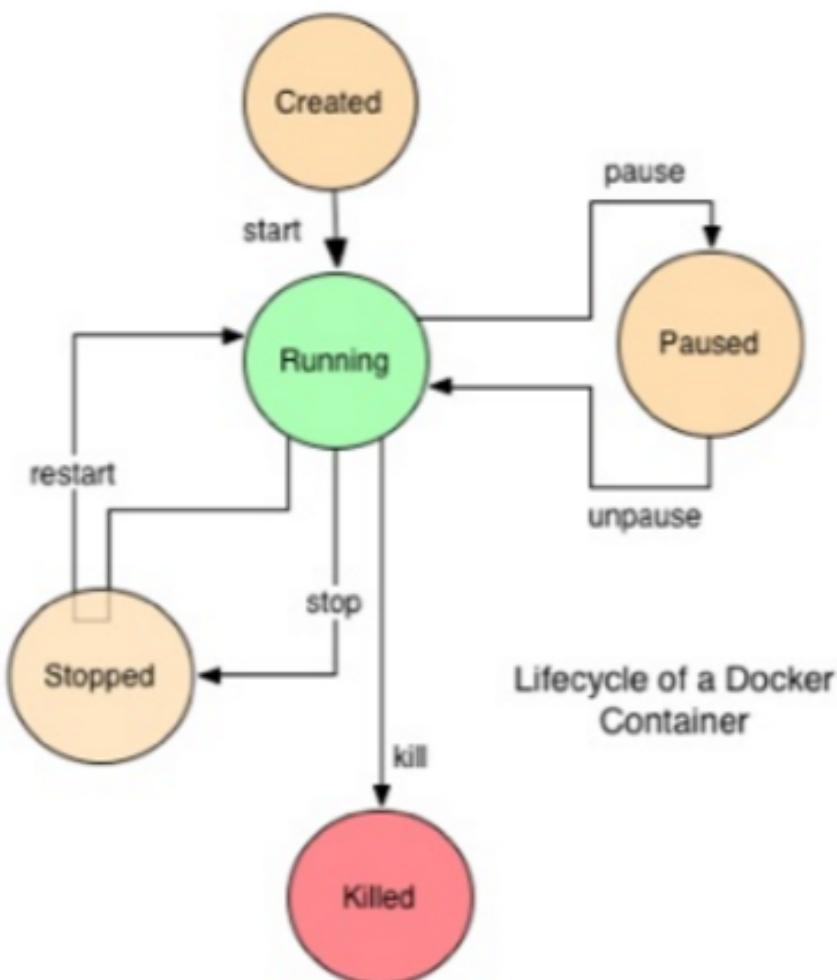
Ubuntu docker image

Docker Container

- *Each container is a fully sandboxed runtime execution environment for our applications.*
- *Containers are launched from Images.*
- *It's a bare boned Linux machine*
- *One process per container (best practice)*
- *The major difference between a container and an image is the top writable layer. All writes to the container that add new or modify existing data are stored in this writable layer. When the container is deleted the writable layer is also deleted. The underlying image remains unchanged.*



Lifecycle of a Container



Docker Hub

The Docker Hub is a public registry/repository that is maintained by Docker Inc. containing a large number of images that you can download and use to build containers.

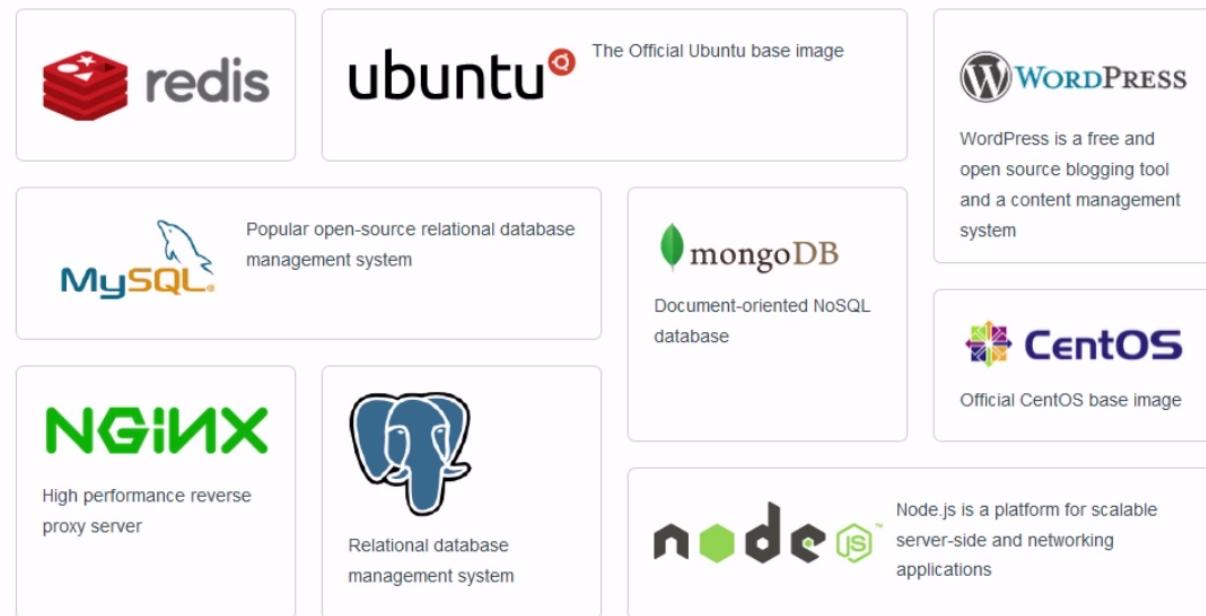
Beginning to work with The Docker Hub is as easy as visiting the site and signing up for an account:

- <http://hub.docker.com>

Once you have an account, you can work with the Hub directly in building and sharing your containers.

- *registry is hub.docker.com*
- *official repos (maintained by companies and docker)*
- *user repos (maintained by users)*

Official Repositories



Docker examples...

```
docker run -i -t centos /bin/bash
```

- ✓ Pulls the centos image
- ✓ Creates a new container
- ✓ Allocates a filesystem and mounts a read-write layer
- ✓ Allocates a network / bridge interface
- ✓ Sets up an IP address
- ✓ Executes a process that you specify
- ✓ Captures and provides application output

Docker examples...

```
docker run -d ubuntu /bin/bash -c “ping 8.8.8.8 -c 30”
```

- ✓ Pulls the ubuntu image if its not there
- ✓ Creates a new container
- ✓ Pings the DNS server (8.8.8.8) 30 times as per the command given
- ✓ Container exits

Physical location of containers on docker host

/var/lib/docker/containers/

```
[root@docker-machine ~]# cd /var/lib/docker/containers/
[root@docker-machine containers]# pwd
/var/lib/docker/containers
[root@docker-machine containers]# ls -l
total 88
drwx----- 2 root root 4096 Oct  5 23:51 37501c6b909bd334c6fca932a406866f9e53129de940a1a787260544071bf7d7
drwx----- 2 root root 4096 Oct  6  02:20 37aedb798f818d8c288d4c9179079ee1764073ccb624e38571a44dc6065dd1
drwx----- 2 root root 4096 Oct  9  07:46 4a7bc8f251e9d485048e86b37d835aebccde32062da52d04131accf594383e09
drwx----- 2 root root 4096 Oct  9  07:35 548121bb44905e6c190157ca1c796639a3079b70e2568e7c8f01c1bca7e5a5a0
drwx----- 2 root root 4096 Sep 30 16:43 6f19eb307b6163e2b937cc3a64620cbff6f7fd1a825e8a17ebd9fbf3e8480842
drwx----- 2 root root 4096 Sep 30 16:52 8430381c71eb028629dd5e4dd3a75ddc6f6f45723ab34b27930fdd9ea6340234
drwx----- 2 root root 4096 Oct  5 17:00 878aebc1b81f68f26c4078f3bf53c530e7fbb8d088cee87c70ada17285bebccb
drwx----- 2 root root 4096 Oct  9  07:46 a267139aeb525bdbd033755a5e12d1a4da3799855ce543719fed48f19145f346
```

Docker CLI

- docker pull 'imagename:version'
- docker images
- docker info
- docker run <options>
- docker attach 'containerid'
- docker stop 'containerid'
- docker restart 'containerid'
- docker commit
- docker inspect
- docker ps <options>
- docker rm 'containerid'
- exit (or "Ctr+P+Q" to keep container running)

Authoring Images with Dockerfile

```
# DOCKER-VERSION 0.10.0

# Pull base image.
FROM ubuntu:14.04

# Install Node.js
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN add-apt-repository -y ppa:chris-lea/node.js
RUN apt-get update
RUN apt-get install -y nodejs
RUN apt-get install -y supervisor
RUN mkdir -p /var/log/supervisor

ADD . /src

# add the supervisor config file to default location that it would look for ADD supervisord.conf /etc/supervisor/conf.d/supervisord.conf

RUN cd /src; npm install

ENV PORT 3001
ENV NODE_ENV development

EXPOSE 3001

CMD ["node", "/src/server.js"]
```

Docker Ecosystem

- Cocaine, Dein, Flynn – Private PaaS
- CoreOS, full distro based on Docker – Linux Distribution
- Red Hat's Project Atomic - Linux Distribution
- Google's Kubernetes – container management
- New Relic Centurian – container management
- OpenStack Integration (In Havana, Nova has Docker driver)
- Vagrant – Docker provider