

Docker

Big Picture



Applications run
businesses

Setting the Scene
Applications run businesses



Applications run
businesses

Applications run on
servers



Applications run
businesses

Applications run on
servers

- Procurement lead times
- Up-front capex
- Ongoing opex



Applications run
businesses

Applications run on
servers

- Procurement lead times
- Up-front capex
- Ongoing opex

Hypervisors allow
multiple apps per
server



\$\$\$\$\$
\$\$\$\$\$



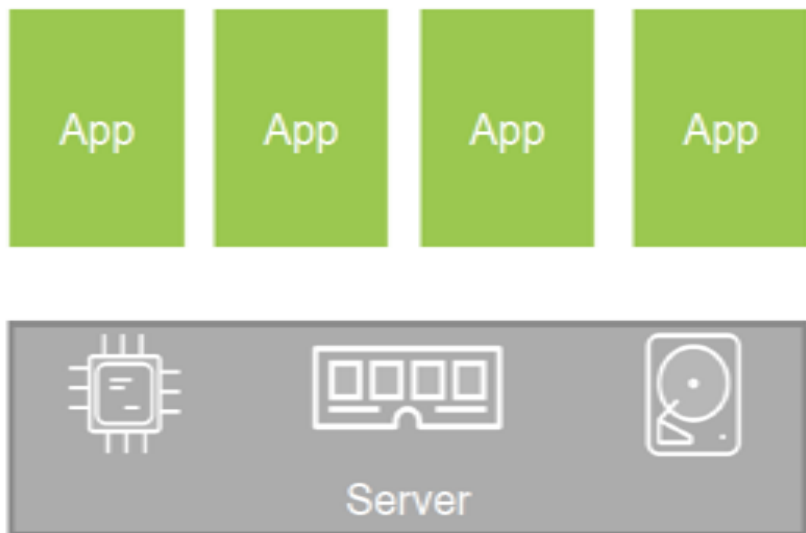
The Trouble with Hypervisors

Applications run
businesses

Applications run on
servers

- Procurement lead times
- Up-front capex
- Ongoing opex

Hypervisors allow
multiple apps per server

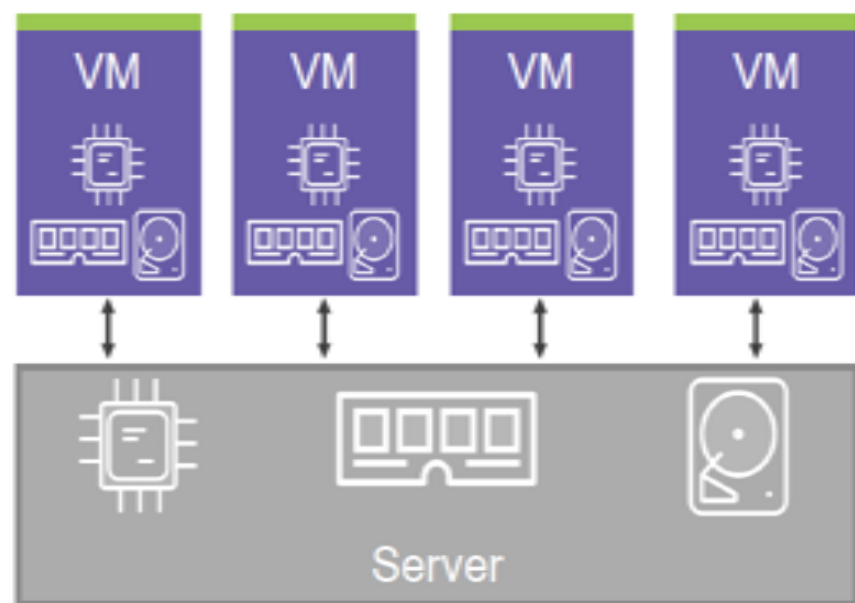


Applications run
businesses

Applications run on
servers

- Procurement lead times
- Up-front capex
- Ongoing opex

Hypervisors allow
multiple apps per server



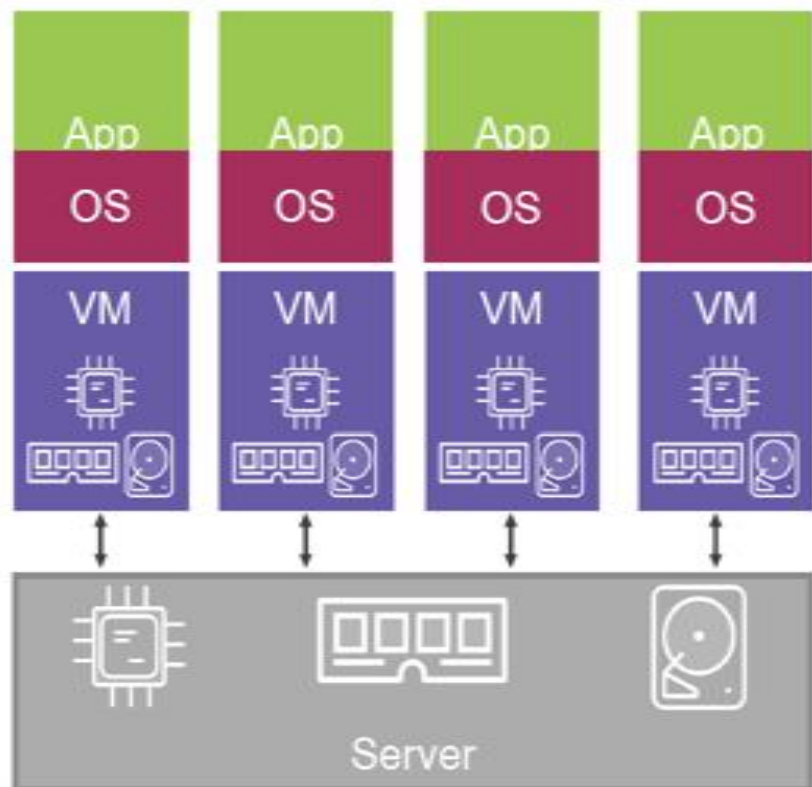
Applications run
businesses

Applications run on
servers

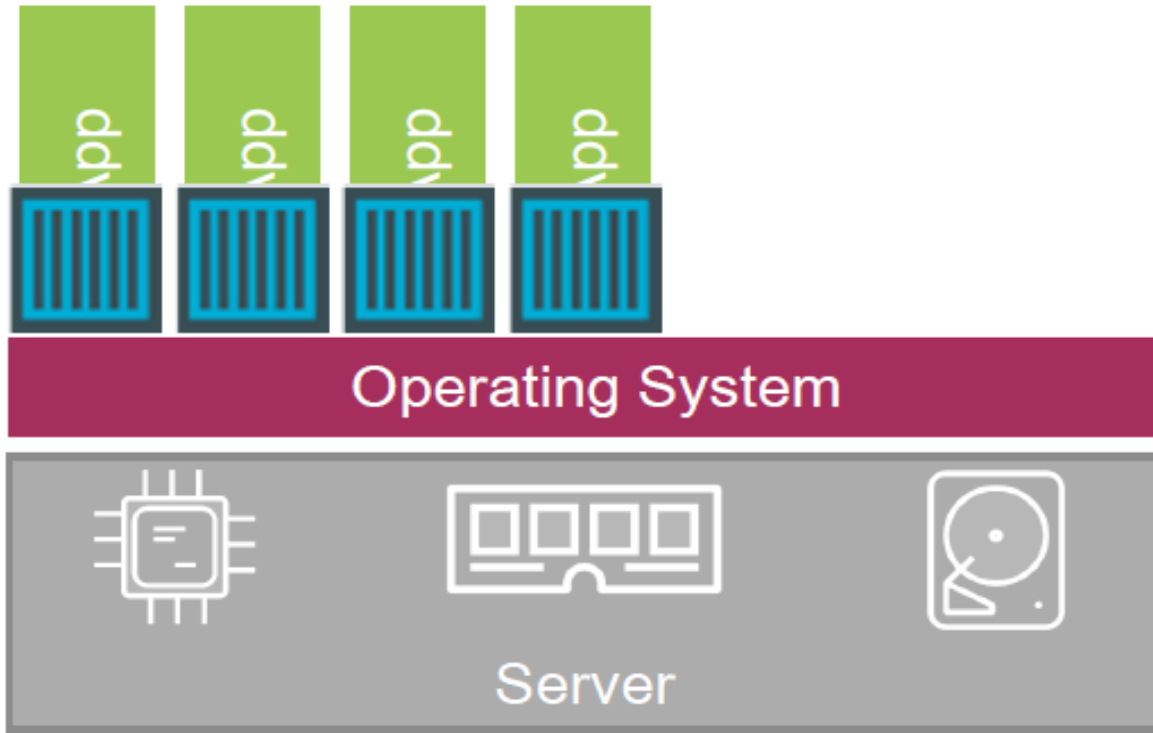
- Procurement lead times
- Up-front capex
- Ongoing opex

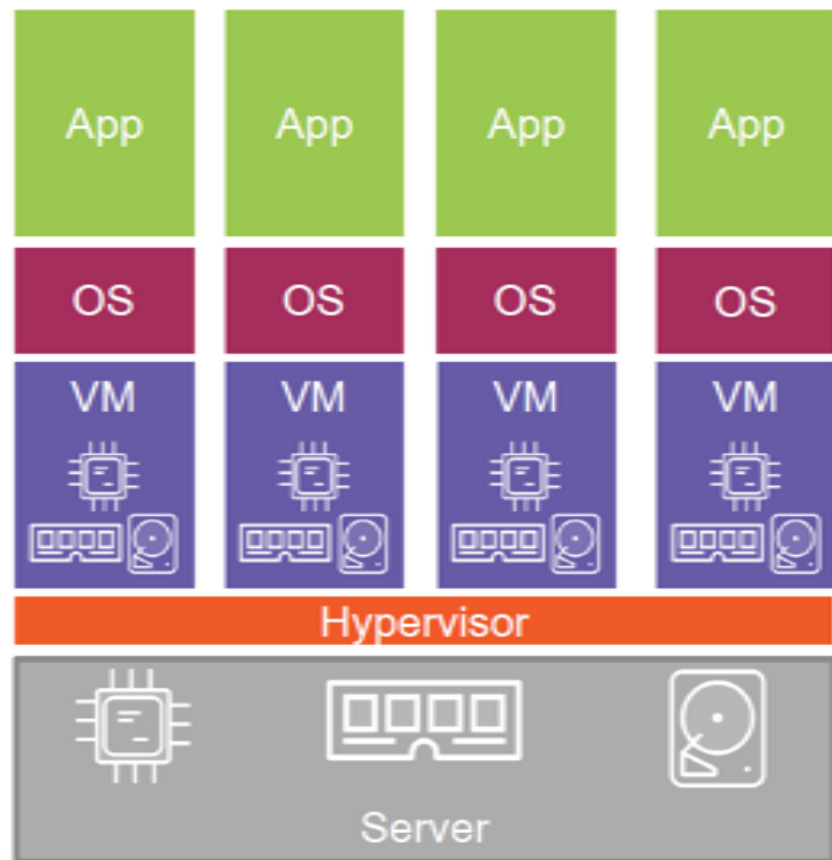
Hypervisors allow
multiple apps per server
Each OS:

- Uses CPU
- Uses RAM
- Uses disk
- May have license cost
- Requires admin time

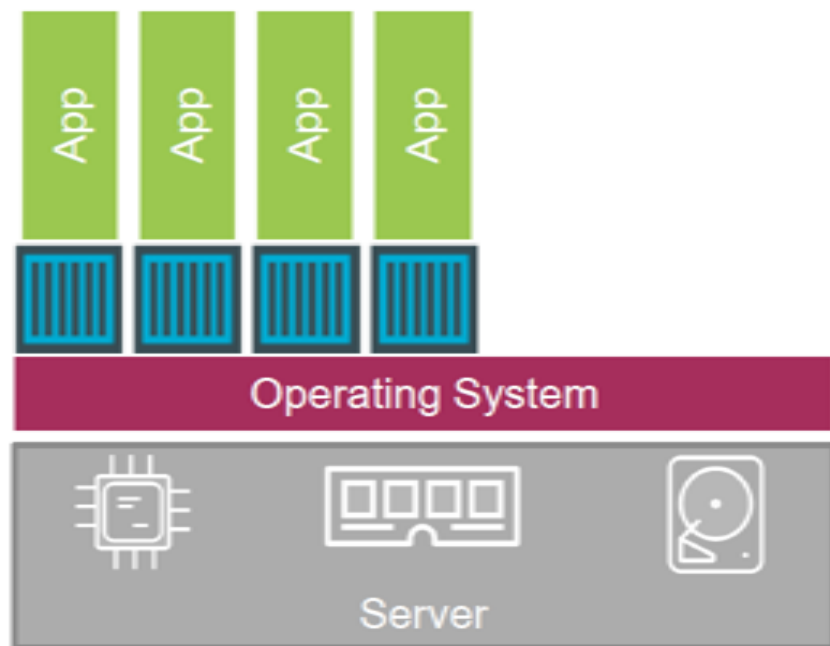


Containers

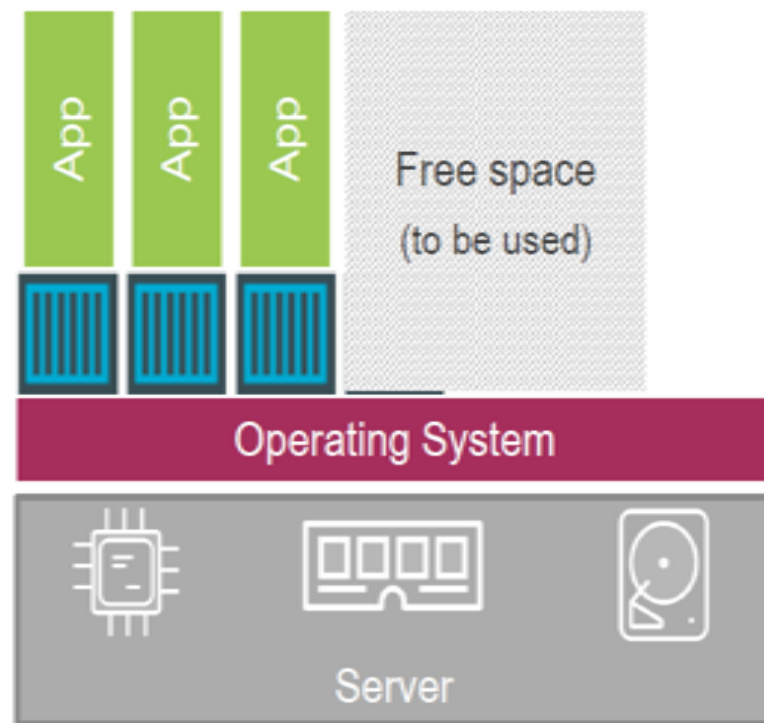
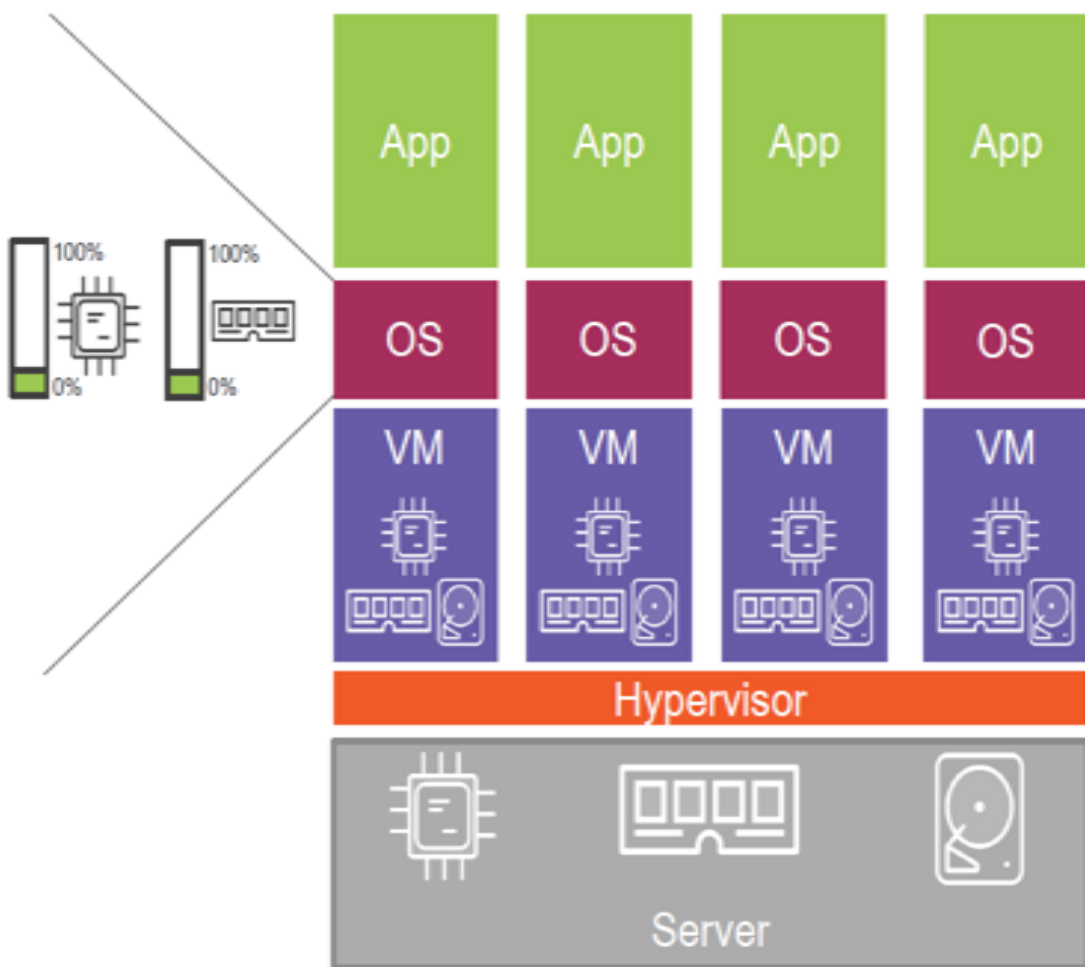




Hypervisor
Architecture



Container
Architecture



Summary

Past
(physical servers)



Wasted!



Wasted!



Wasted!

Present
(Hypervisor virtualization)



More efficient than
physical servers

Could be better!

Present/Future
(Containers)



More efficient than
Hypervisor
virtualization

Virtualization 2.0

Less mature than
Hypervisor virtualization

Less mature than
Hypervisor ecosystem

Docker Inc.



Silicon valley startup

Raised \$180M

Made several
acquisitions



The Docker Project

Open source

- Apache license 2.0

>240K repositories

Billions of downloaded
images

Support from major
industry leaders

The Open Container Initiative

Lightweight governance

Open standards



What Kind of Work Will Containers Do

Stateless: Does not keep any changes or data

Stateful: Keeps changes and data

Stateful vs Stateless

Q: Are containers just for stateless apps and services?

A: Docker and containers can handle stateful and stateless workloads

Containers are great at modern cloud-native apps

Far superior to Virtual Machines at this

Containers and Traditional Apps

Containers and Traditional Apps



docker

A lot of early messaging from Docker Inc. was about using containers for modern cloud-native apps

- This is where containers naturally excel
- It does not mean that containers cannot be used for traditional enterprise apps

Stateful or Stateless

Docker and containers
work with both types

It's common to have both

Stateful

The Docker storage
backend is now pluggable

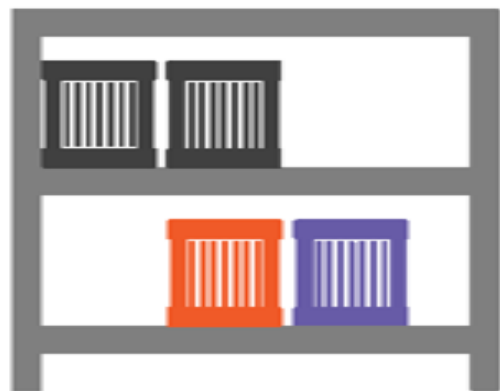
- Has lead to lots of plugins
supporting portability of
stateful containers (and
their data)

Places to store and retrieve container images

Docker Hub is the official Docker registry

3rd party registries exist

Container Registries



Places to store and retrieve container images

Docker Hub is the official Docker registry

3rd party registries exist

Registries contain repositories

Repositories can be public or private

Container Registries



Registry Security

Are public repositories wide open to the world?

- Yes for pulling (downloading)
- No for pushing (uploading changes)

Private repositories are not open to the world

Private registries are available

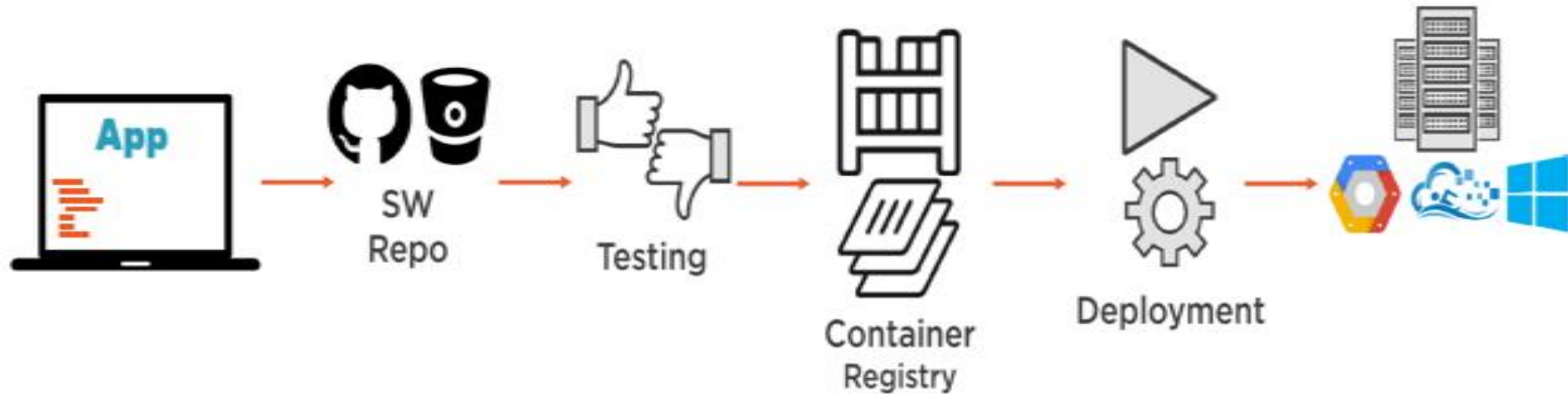
- Run inside of your corporate network
- Docker Trusted Registry (DTR)
- Quay Enterprise
- etc.

~~Docker~~ Content



Container registries are becoming central to application infrastructure and application delivery

Automated Workflow



Summary

Registries are the
enterprise IT App Store



Lots of registries exist



Registries contain
repositories



Private Registries exist

