Docker containers and/or/vs virtualization Overview

Tomasz Jordan Kruk, Ph.D.
Warsaw University of Technology
www.pw.edu.pl

Agenda

1. Docker for virtualization admin
2. Docker - application delivery technology
3. Synergy between VMs and Docker containers
4. Docker deployment – context aspects to consider

Presentation based on material from document
Docker for the Virtualization Admin
at docker.com

Docker for virtualization admin

- Developers build a **Docker image** that includes exactly what they need to run their application: starting with the basics and adding in only what is needed by the application

- **Virtual machines** start with a full operating system and, depending on the application, developers may or may not be able to strip out unwanted components

Docker - application delivery technology (1/2)

Docker is not a virtualization technology, it's an application delivery technology:

- with Docker containers the abstraction is the application; or more accurately a service that helps to make up the application,
- how does a sysadmin backup a Docker container - they don't,
- the application data doesn't live in the container, it lives in a Docker volume that is shared between 1-N containers as defined by the application architecture. Sysadmins backup the data volume, and forget about the container,
- optimally Docker containers are completely stateless and immutable.

Docker - application delivery technology (2/2)

Docker is not a virtualization technology, it's an application delivery technology:

- patches aren't applied to running Docker containers.
- admins update their existing Docker image, stop their running containers, and start up new ones.
- a container can be spun up in a fraction off a second, these updates are done in exponentially more quickly than they are with virtual machines.
- applications in Docker have much better horizontal scaling.

Synergy between VMs and Docker containers

- running your application in a set of Docker containers doesn't preclude it from talking to the services running in a VM.

- capacity optimization:
  - by mixing and matching Docker hosts with traditional VMs, sysadmins are getting **the maximum utilization out of their physical hardware**.

Agility, portability, control

Important feature of Docker - flexibility it affords IT organizations. The decision of where to run your applications 100% based only on what's right for your business.

Docker hosts could run on vSphere, Azure, AWS, physical servers. Docker containers deliver great combination of

- agility
- portability
- control

Docker containers – where/on what to run

Would container be better served running on virtual machines or bare metal physical servers?

- choice based solely on what's right for application and business goals
  - physical or virtual,
  - cloud or on premise.

- mix and match as your application and business needs do dictate.

Mix and match environments

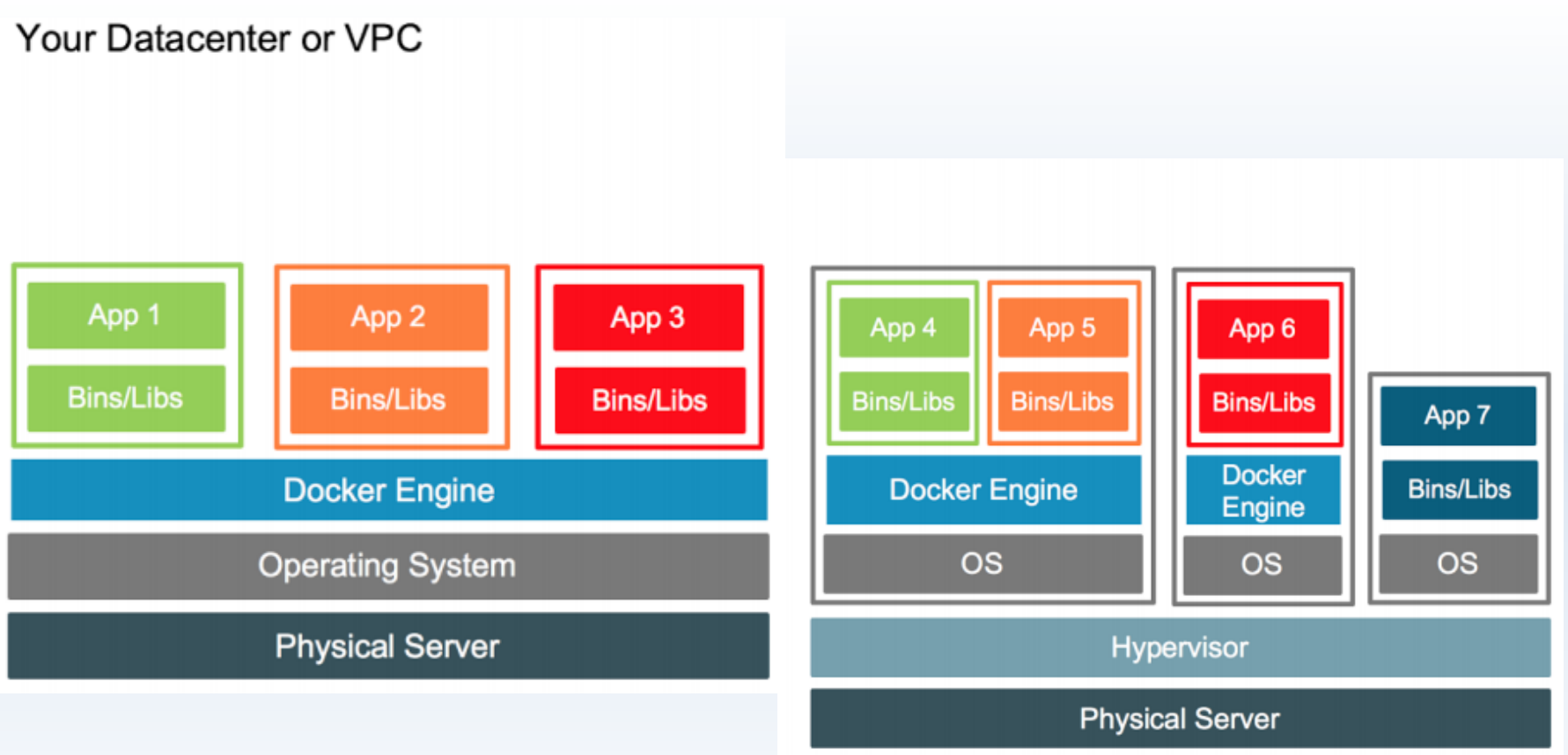

Figure from *Docker for the Virtualization Admin* at docker.com

Docker deployment – context aspects to consider

Bare metal or virtual?

- latency,
- capacity,
- mixed workloads (i.e. Windows and Linux),
- disaster recovery,
- existing investments and automation frameworks,
- multitenancy (forced additional isolation with different OS kernels),
- resource pools / quotas,
- automation/APIs (automation rule out bare metal),
- licensing costs (hypervisor costs).