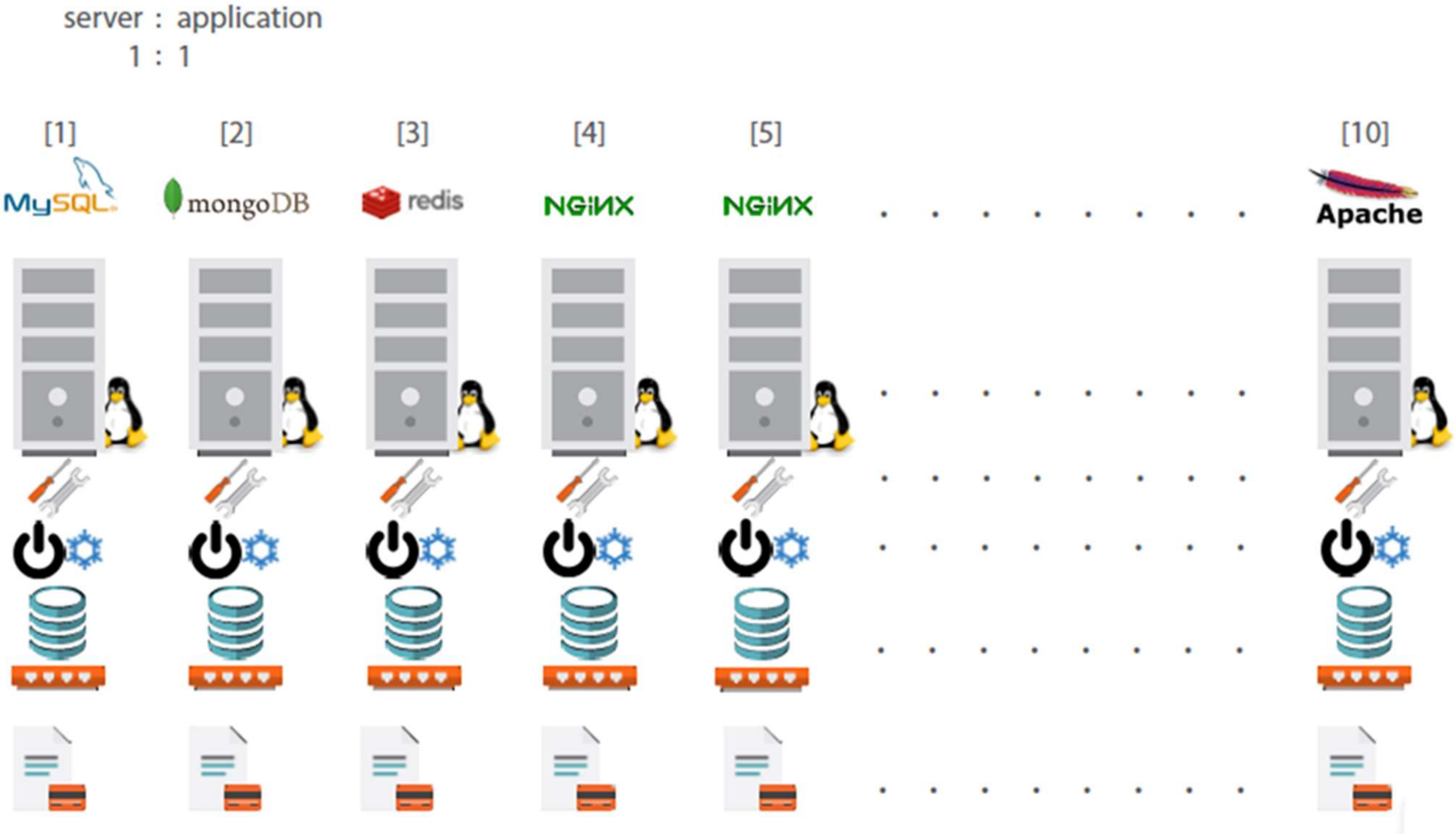
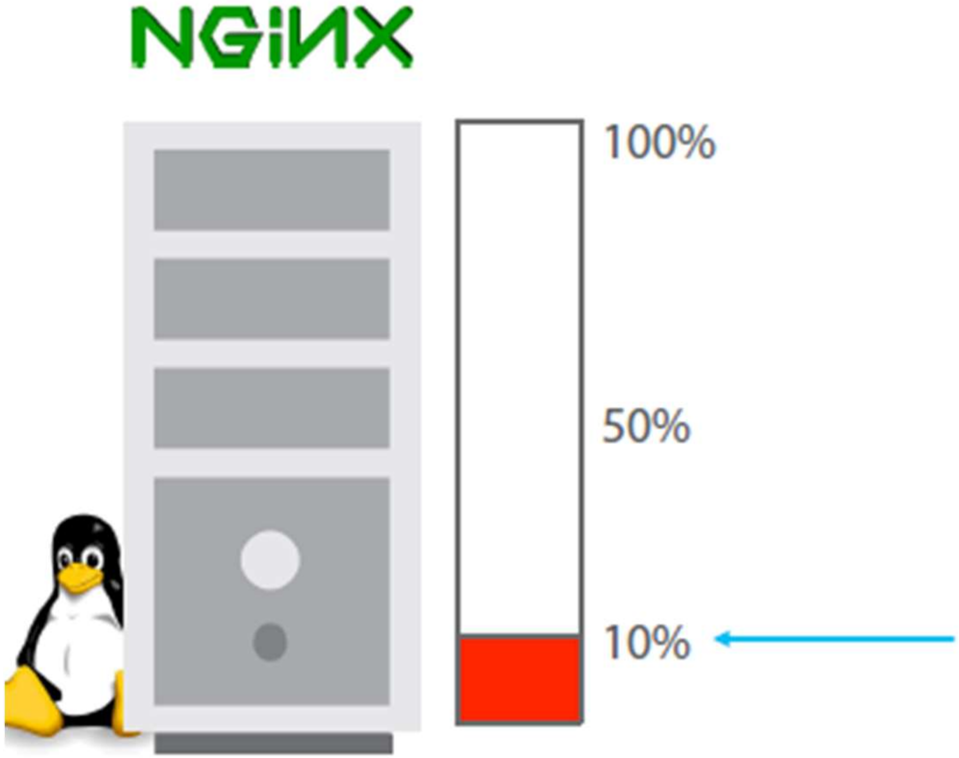
Docker Essentials

Docker

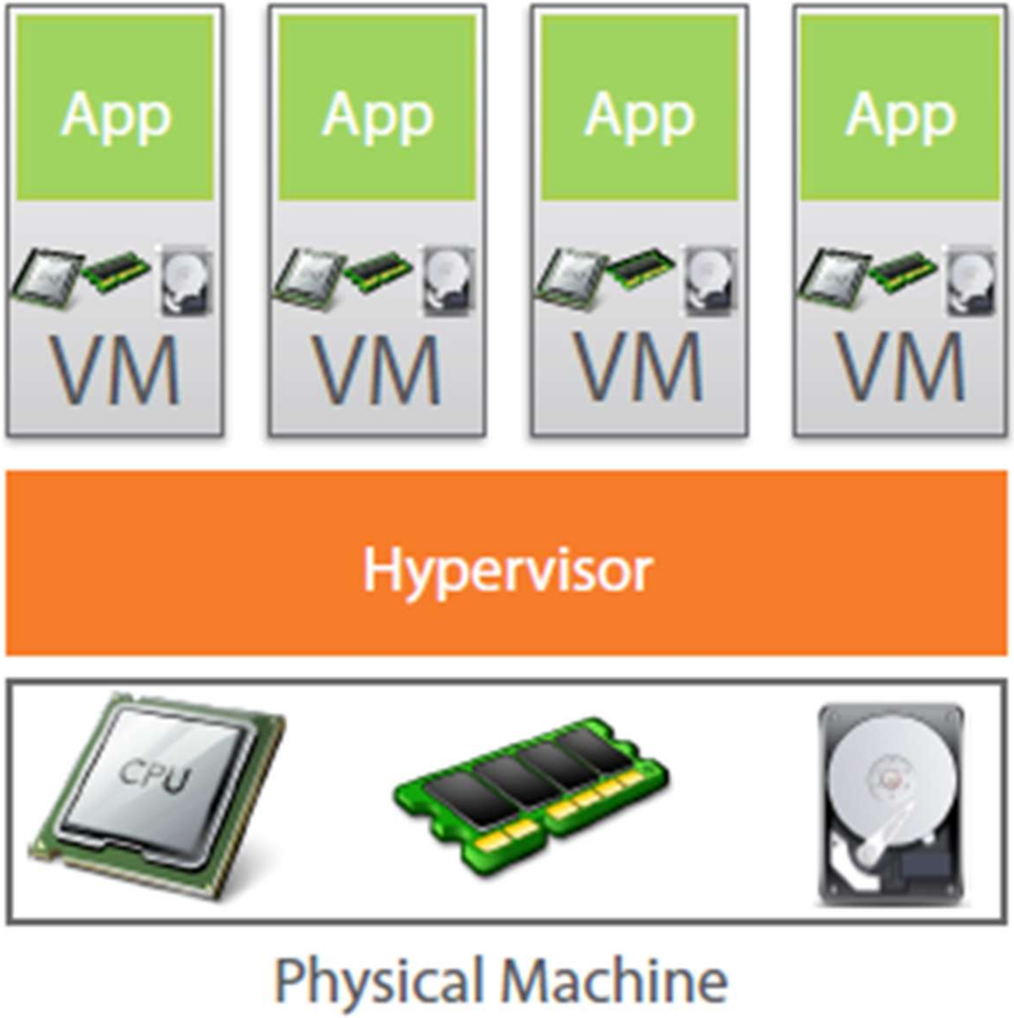
Traditional Deployment Architecture



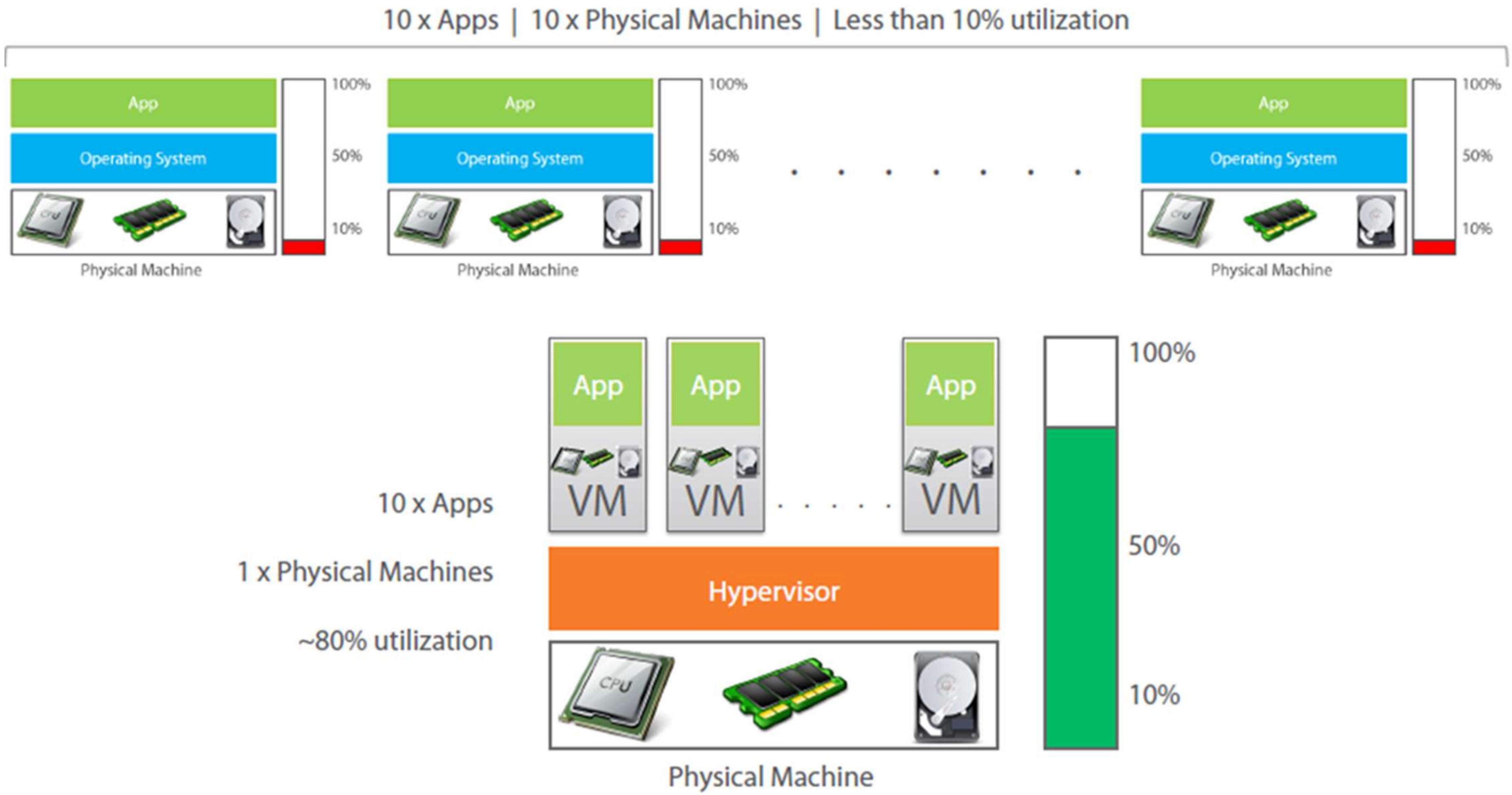
Less Utilization in Traditional Architecture



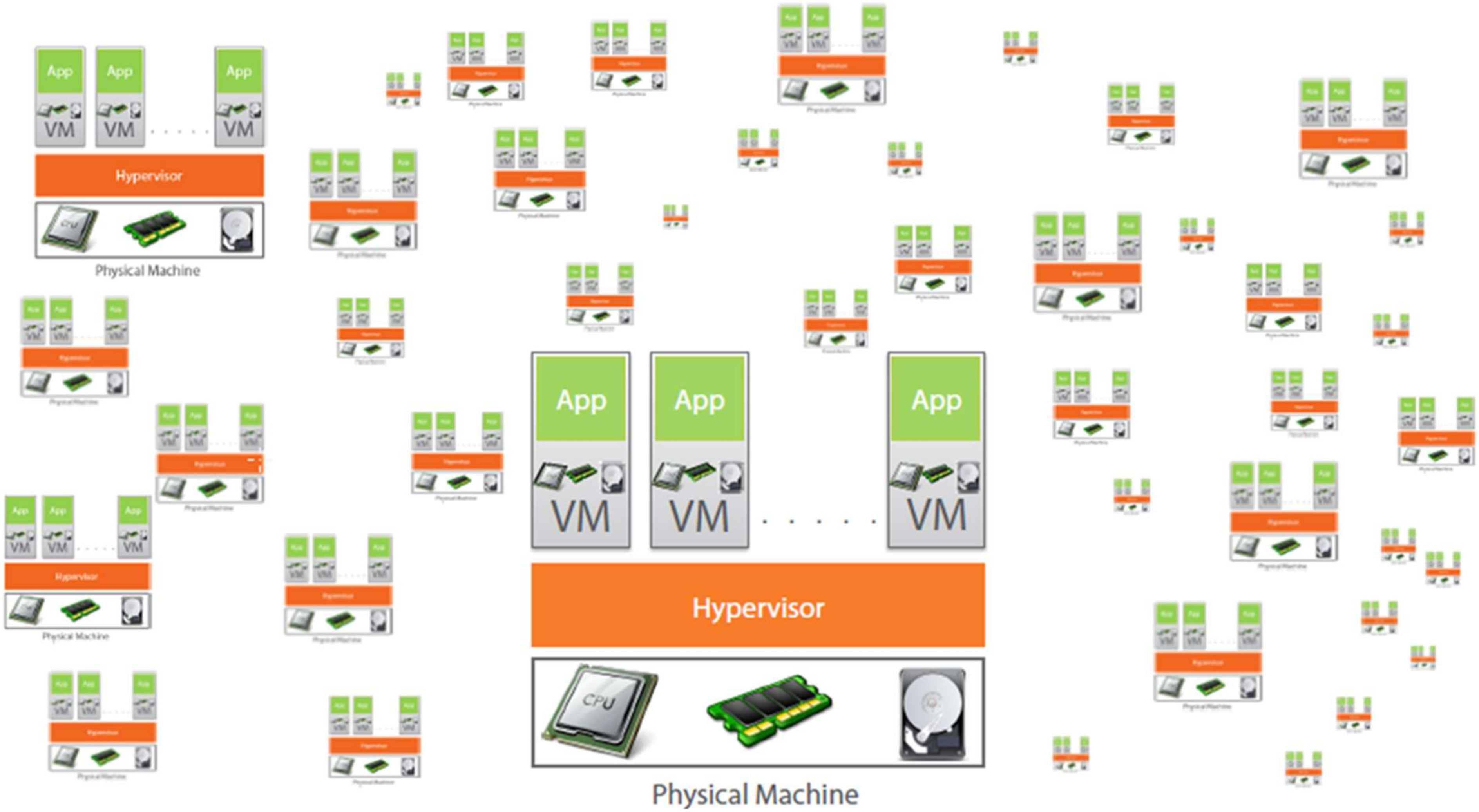
Virtual Machine to the Rescue



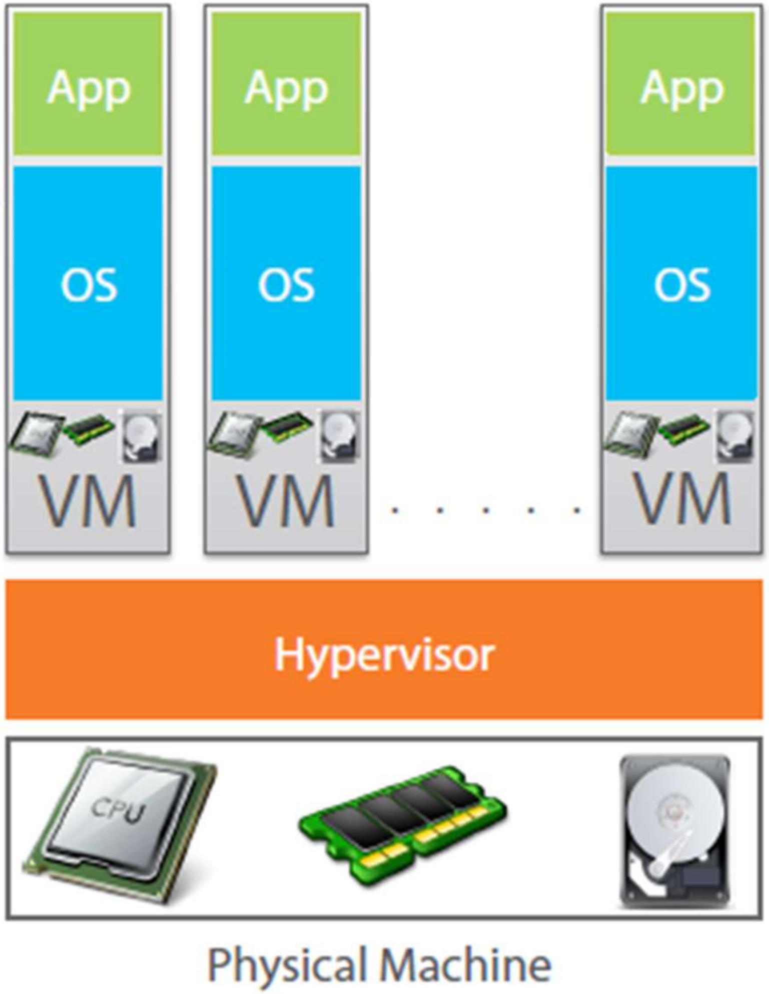
Virtual Machine provides better utilization



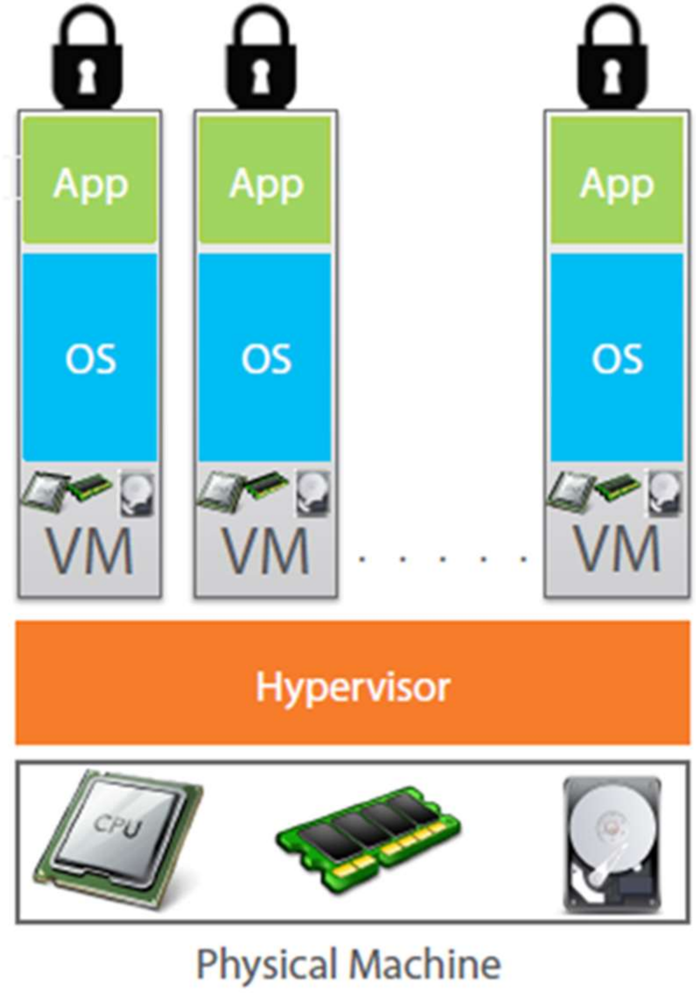
But Virtual Machine increases Licensing Cost



Each VM needs a separate OS

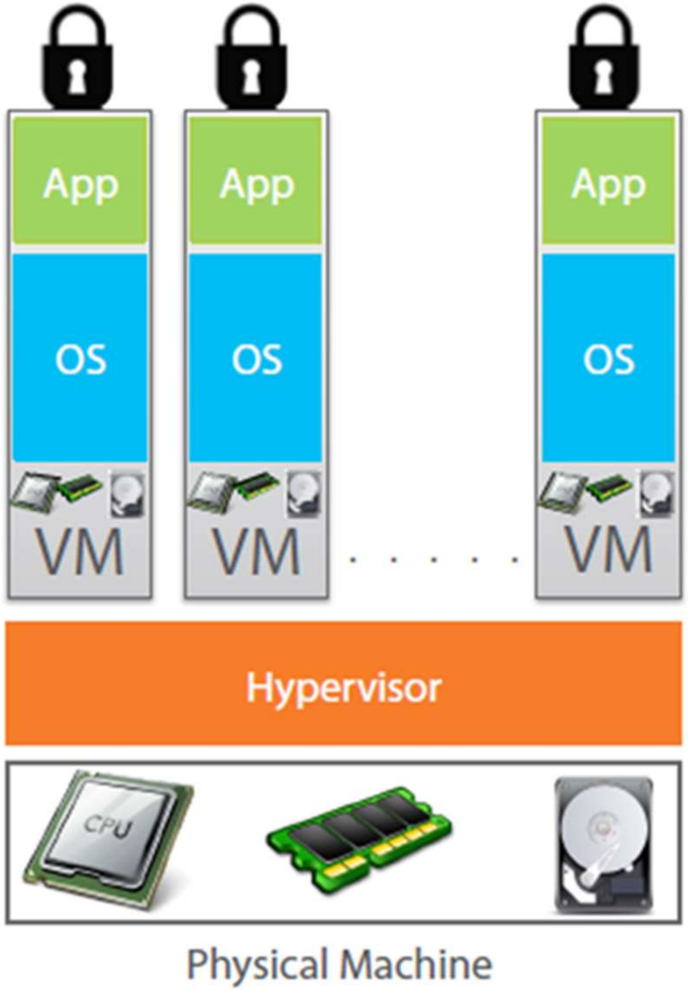


More OSes doesn’t increase Business Value



> OS != Business Value

OS takes most of the Resources

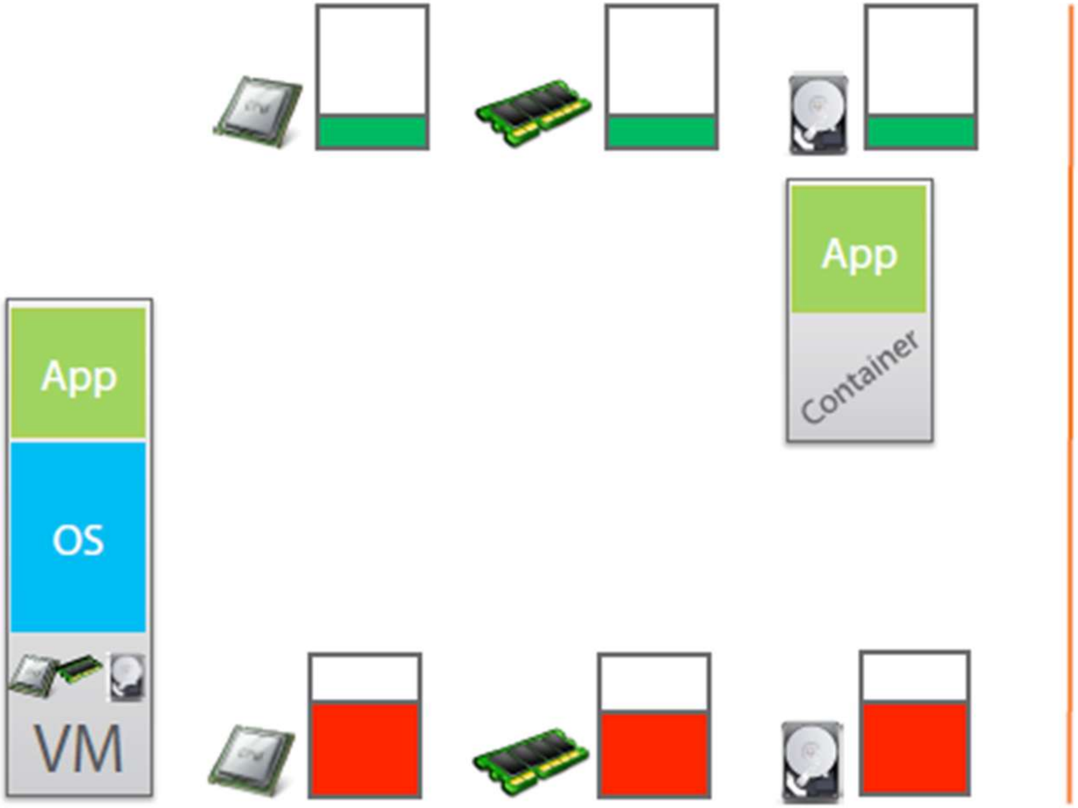


Why use separate OS for each App?

Containerization

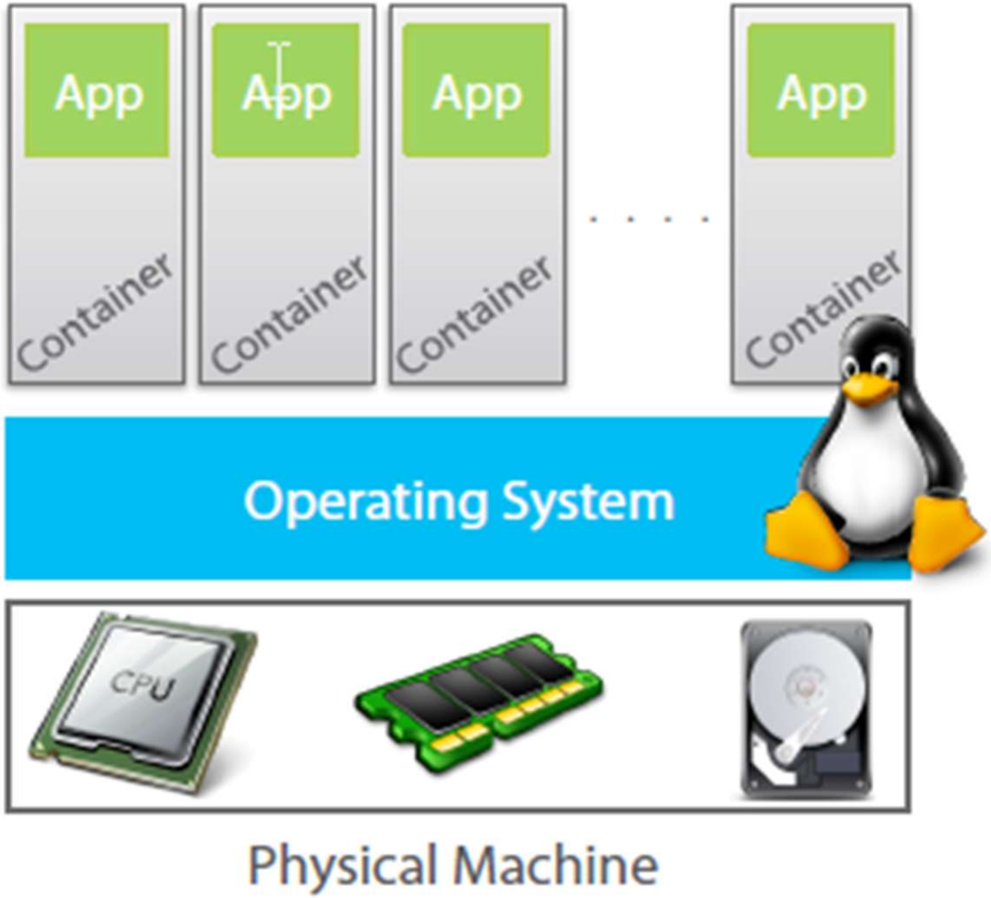
- Encapsulation of an application and its required environment.
- The process of packaging an application along with its required libraries, frameworks, and configuration files together so that it can be run in various computing environments efficiently.

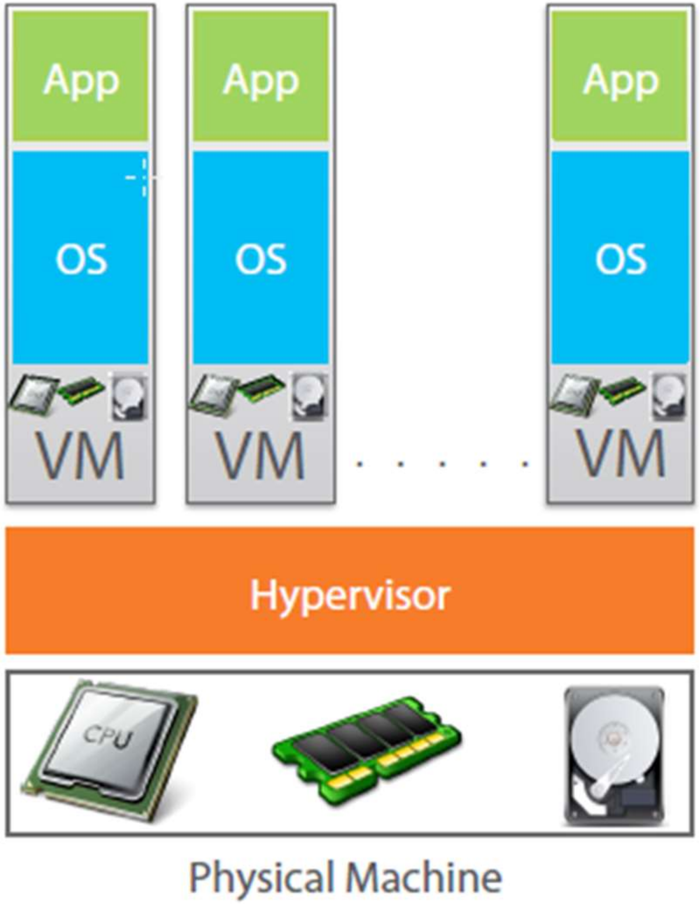
Containers to the Rescue



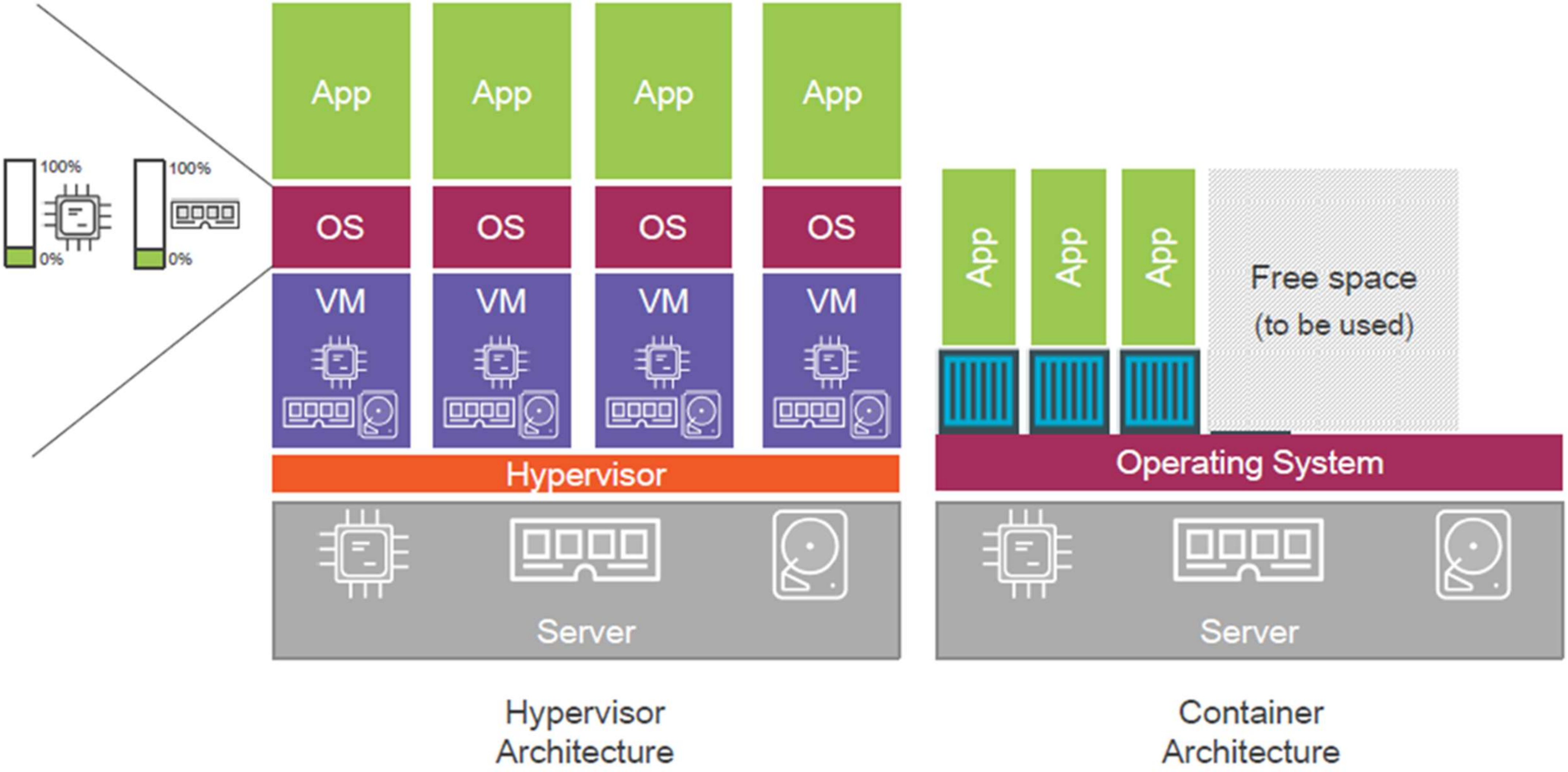
Containers are more lightweight than Virtual Machines

Containers vs VM

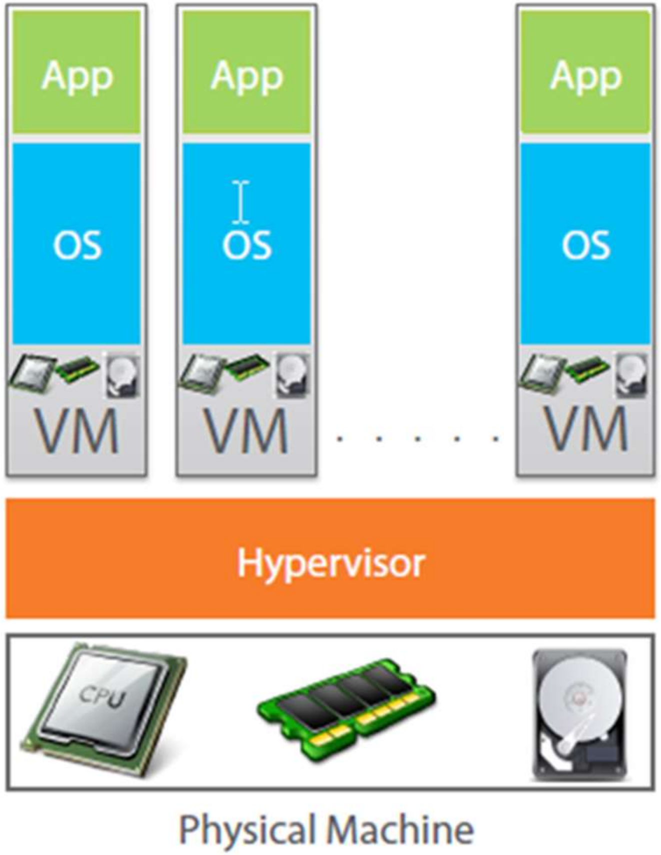


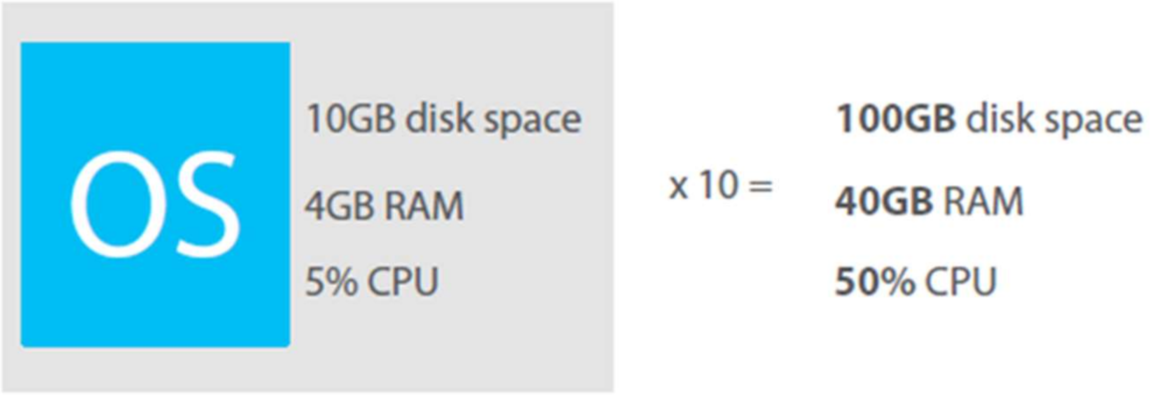


Containers vs VM



OS takes more resources and Licensing cost





10GB disk space

4GB RAM

5% CPU

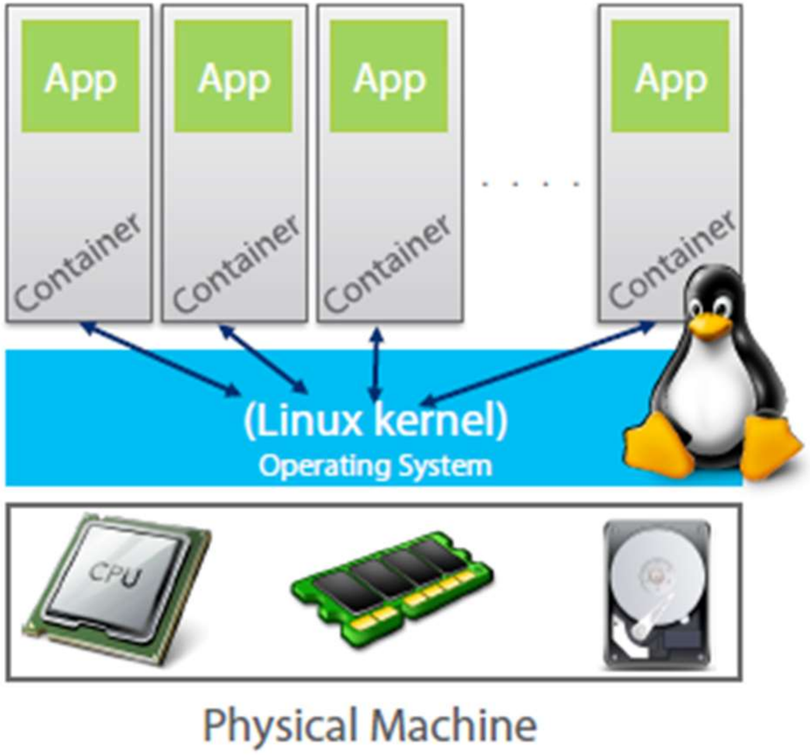
x 10 =

100GB disk space

40GB RAM

50% CPU

Containers takes less resources



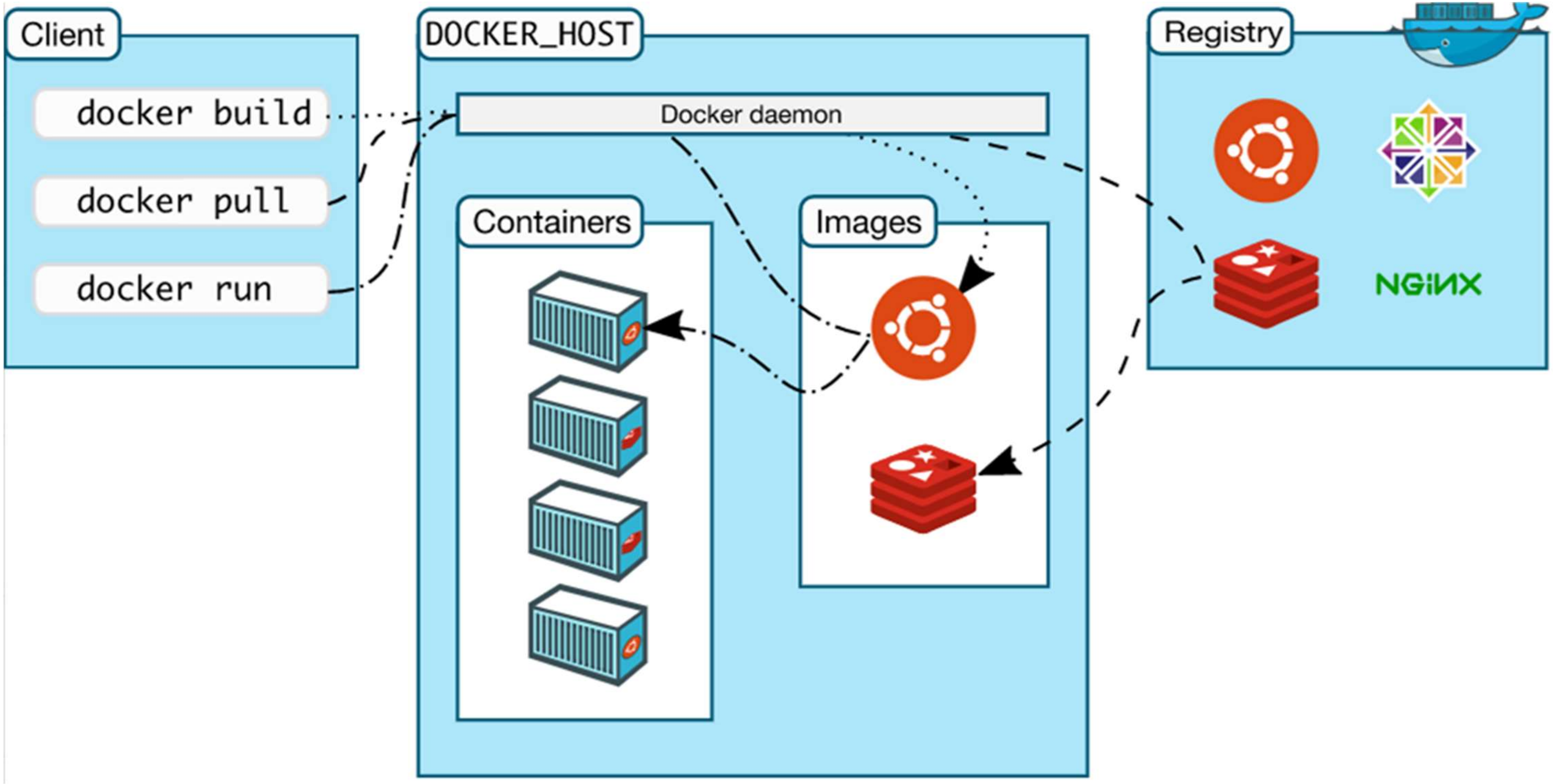
Containers consume less CPU, RAM and disk resource than Virtual Machines

What is Docker?

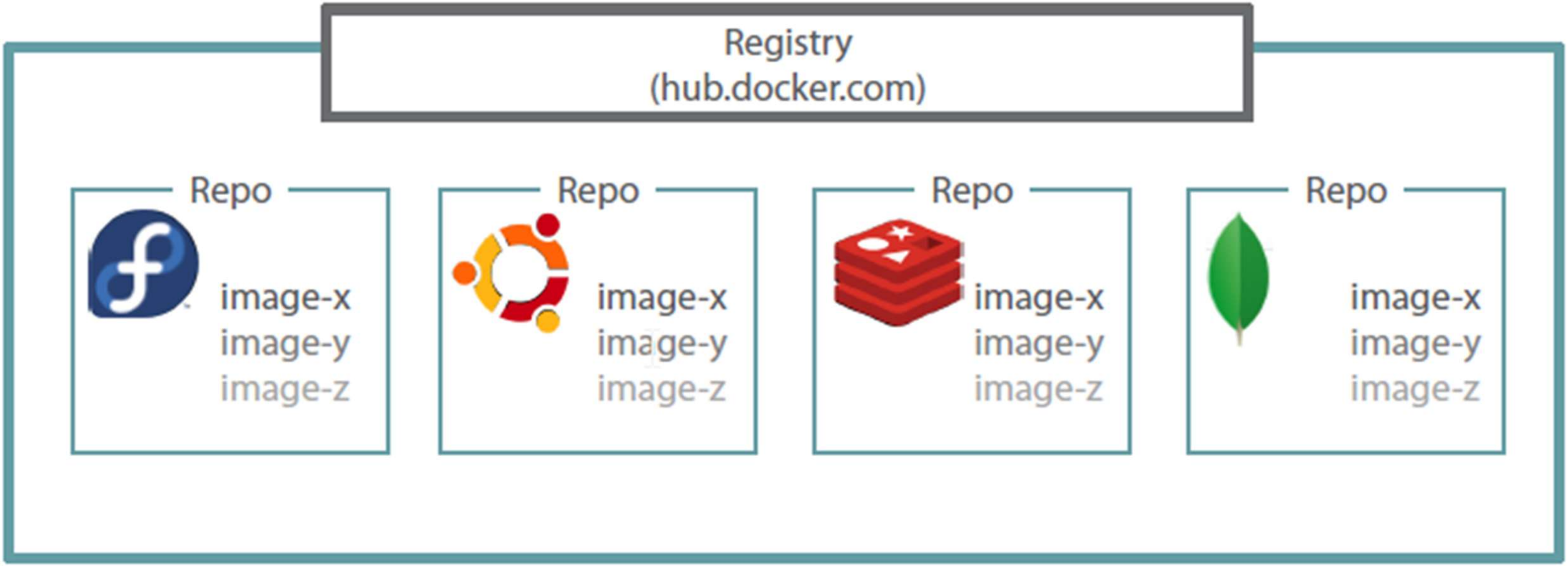
- Docker is an open-source project
  - that automates the deployment of applications inside software containers,
  - by providing an additional layer of abstraction and
  - automation of operating system–level virtualization on Linux.

Practical

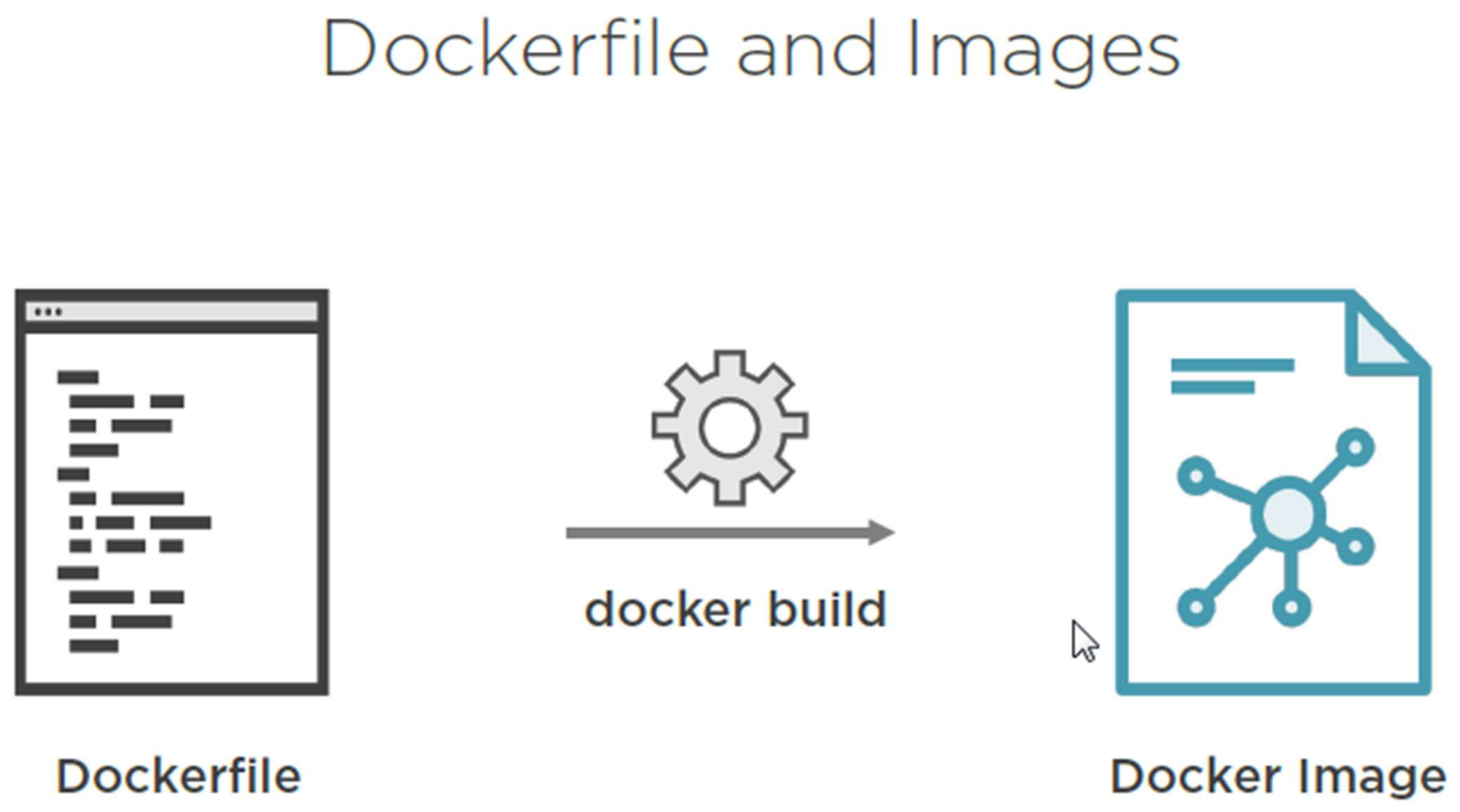
Docker Architecture



Docker Registry



Dockerfile



Dockerfile and Images

Thanks