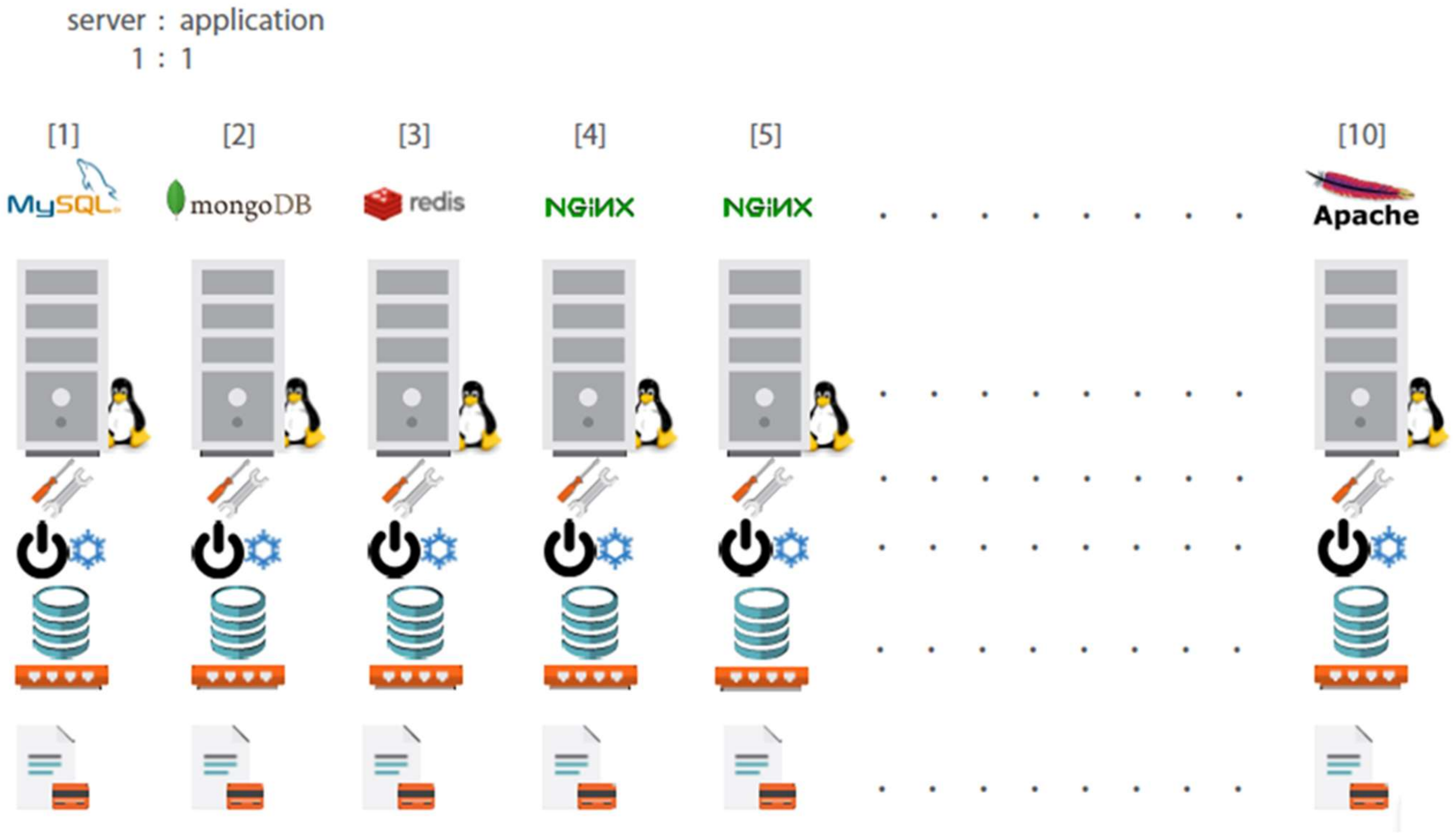
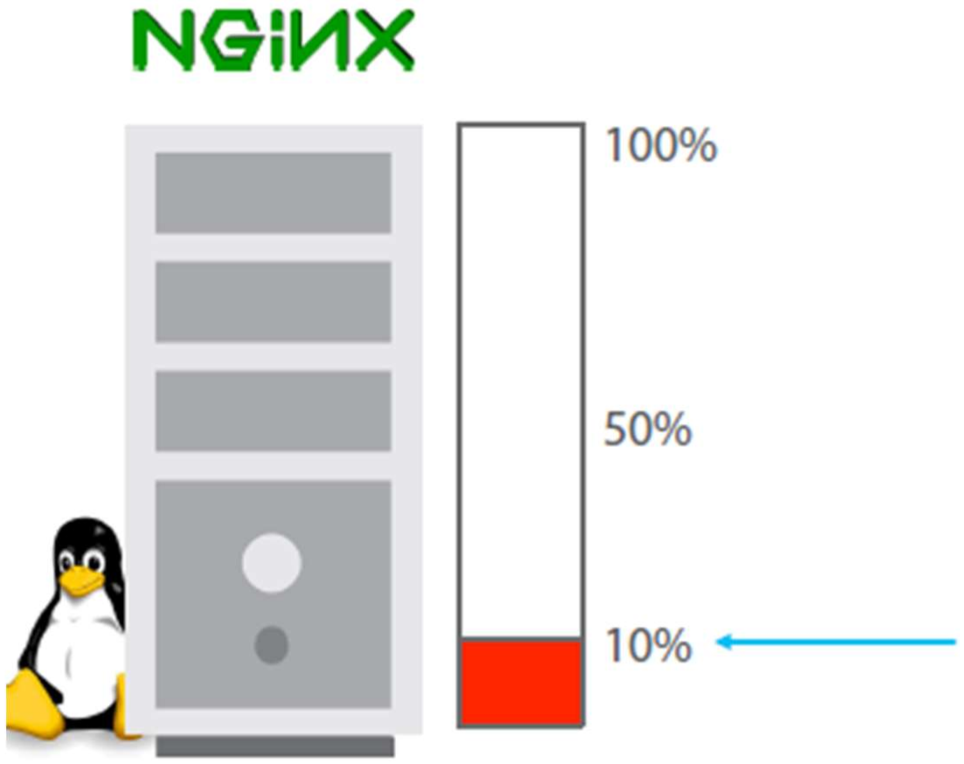
Docker

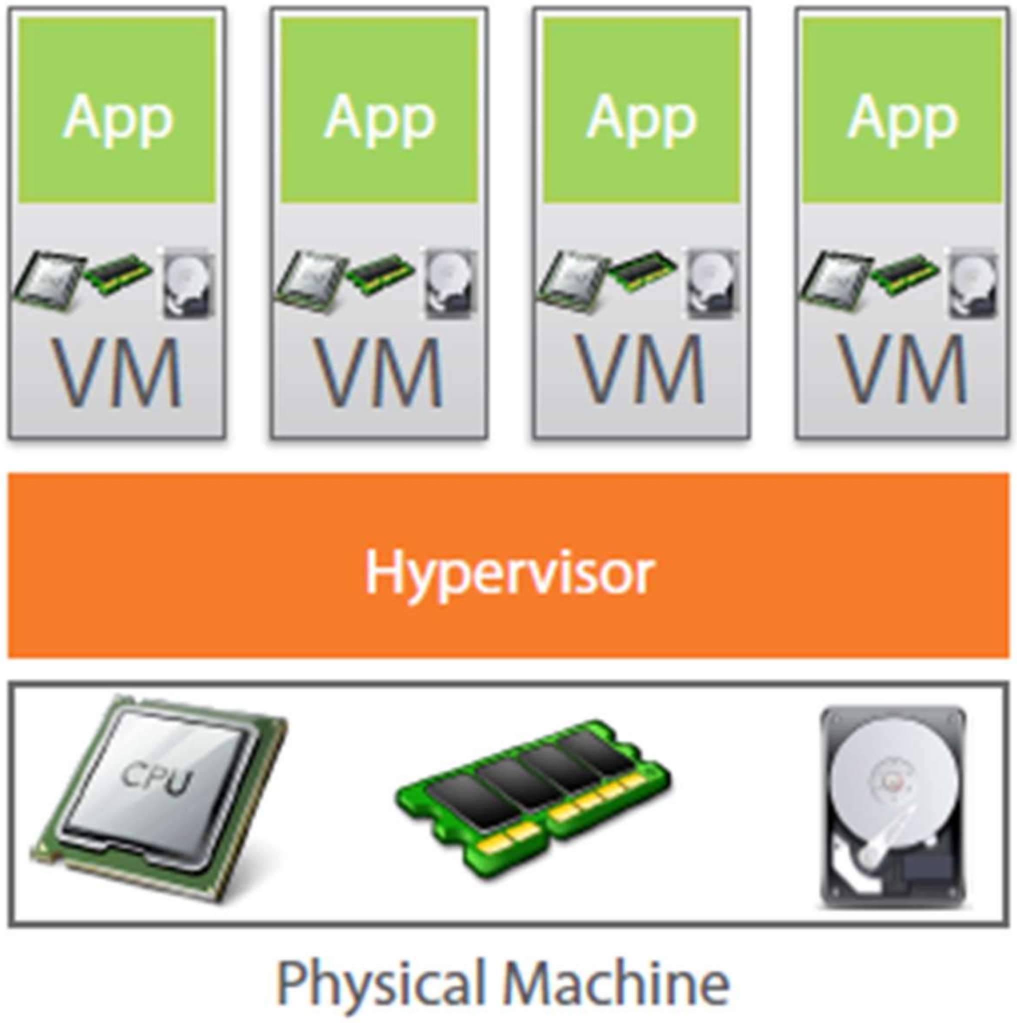
Traditional Deployment Architecture



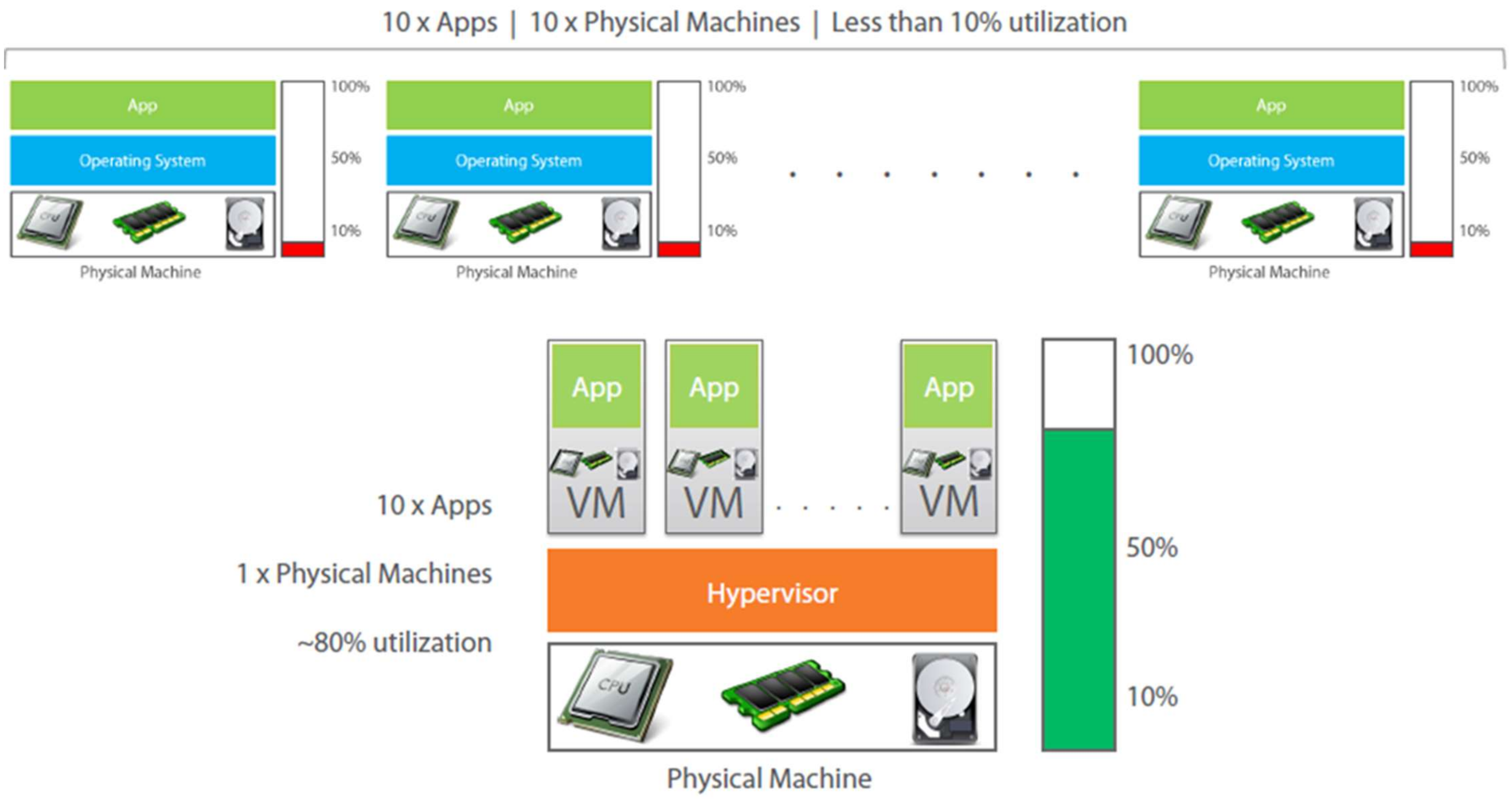
Less Utilization in Traditional Architecture



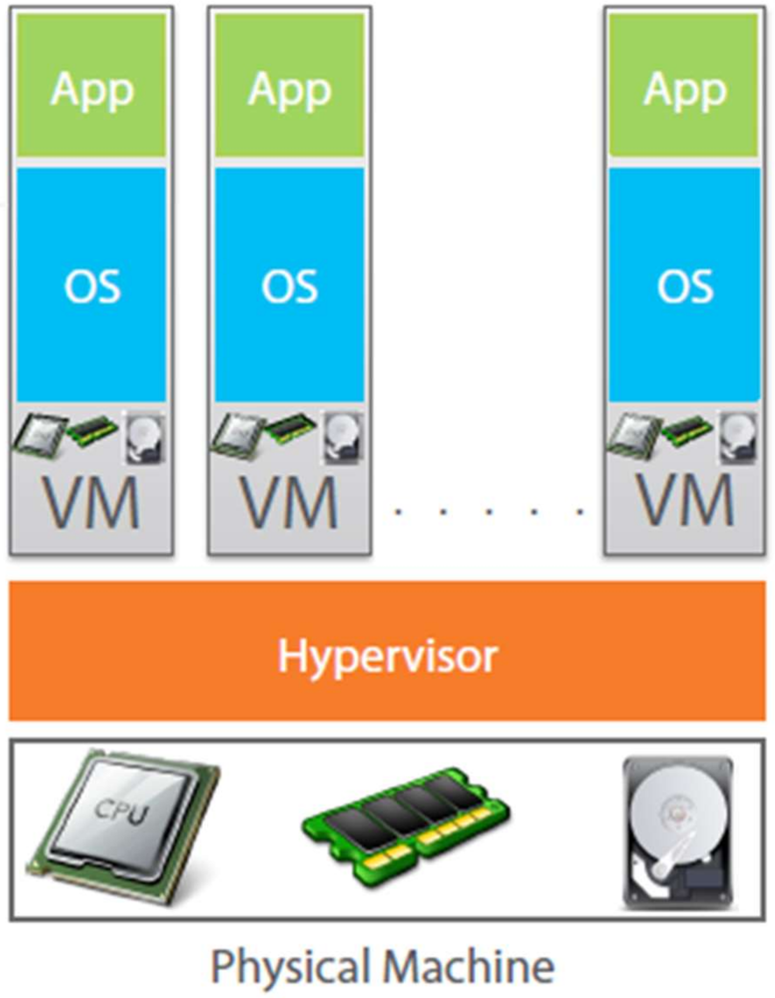
Virtual Machine to the Rescue



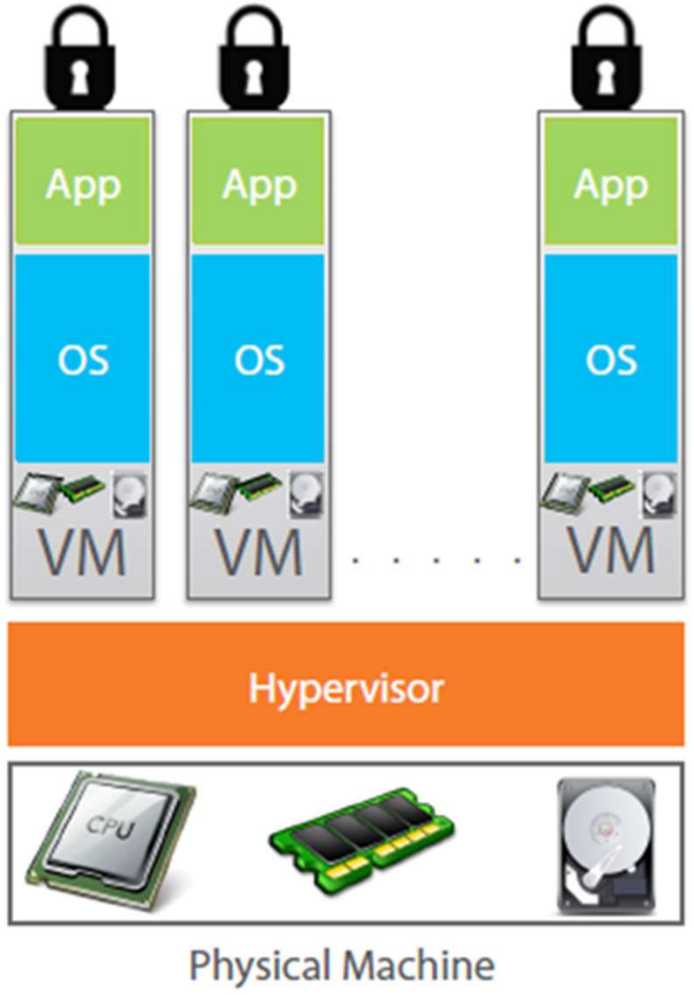
Virtual Machine provides better utilization



Each VM needs a separate OS

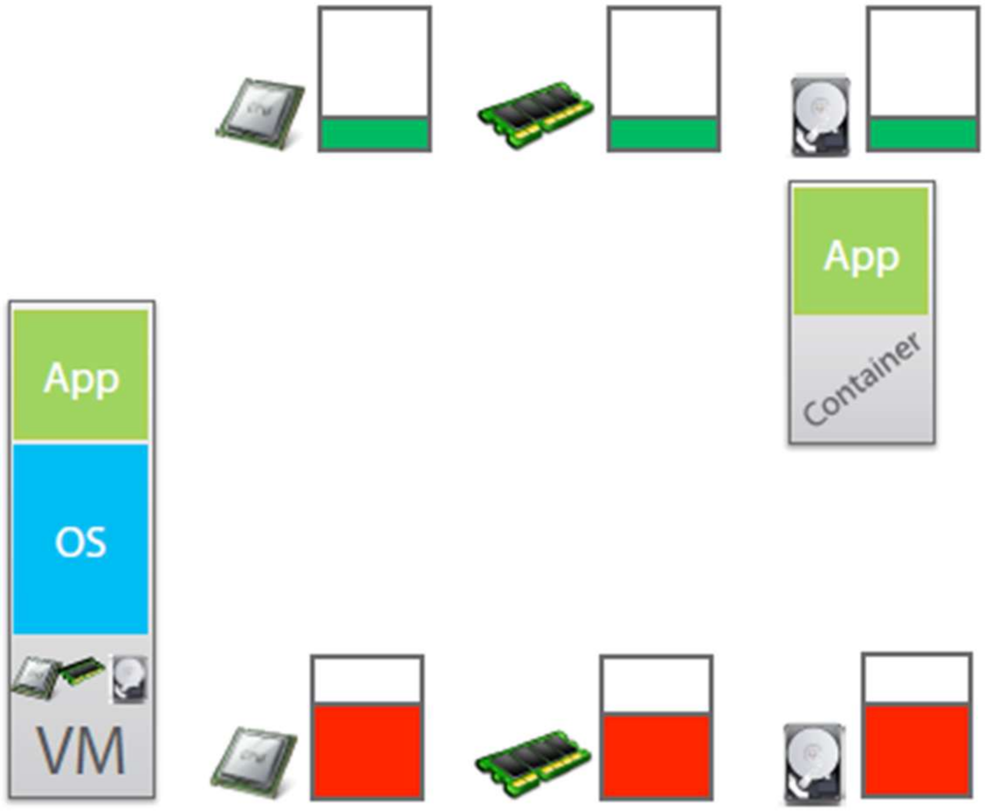


OS takes most of the Resources



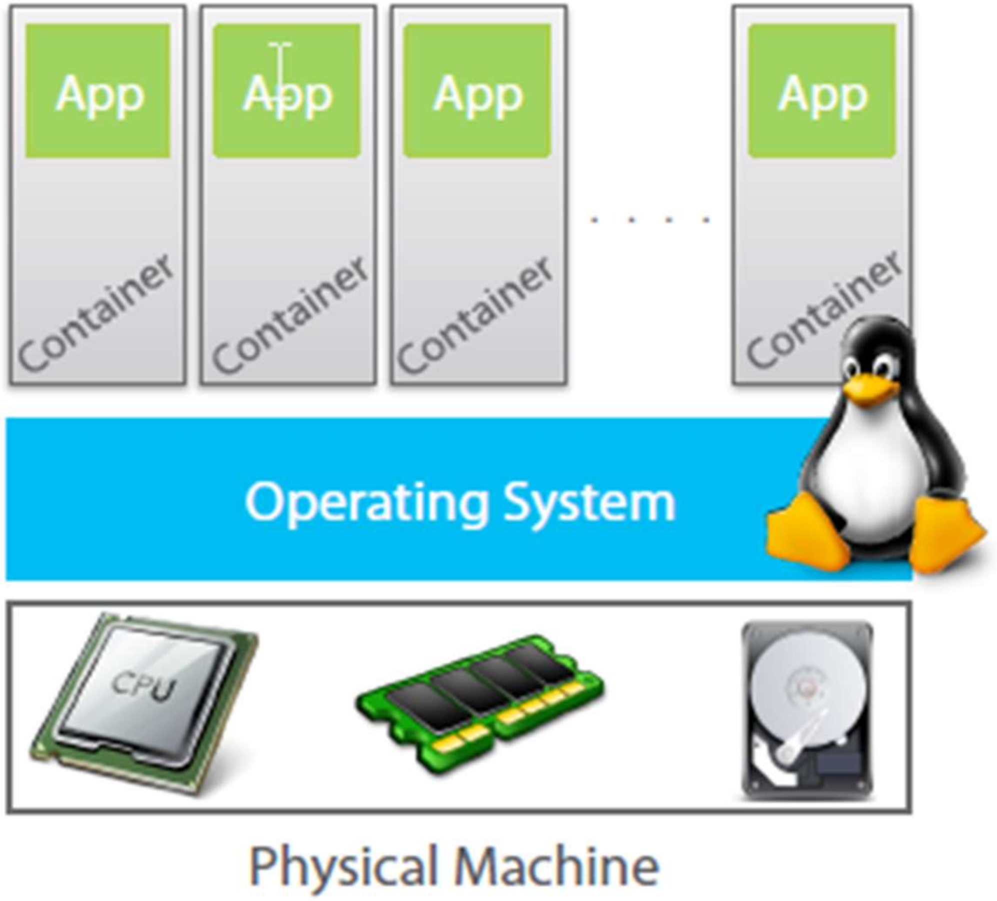
Why use separate OS for each App?

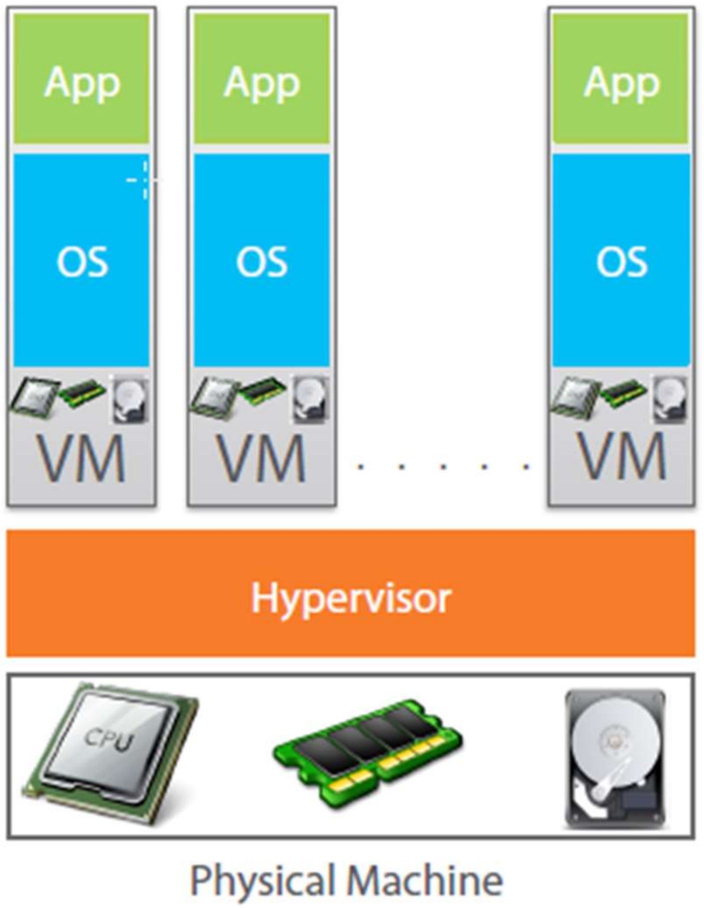
Containers to the Rescue



Containers are more lightweight than Virtual Machines

Containers vs VM

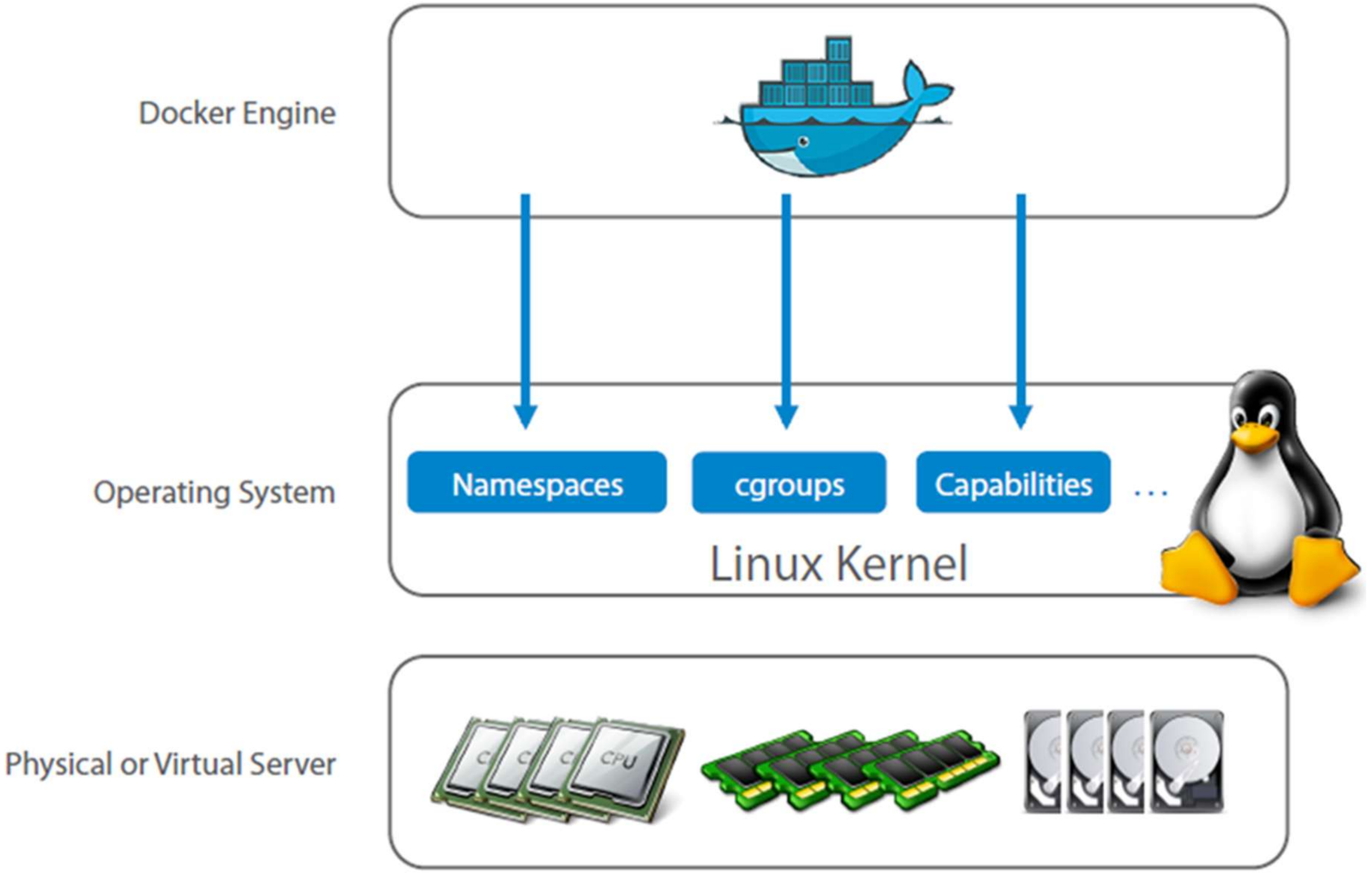




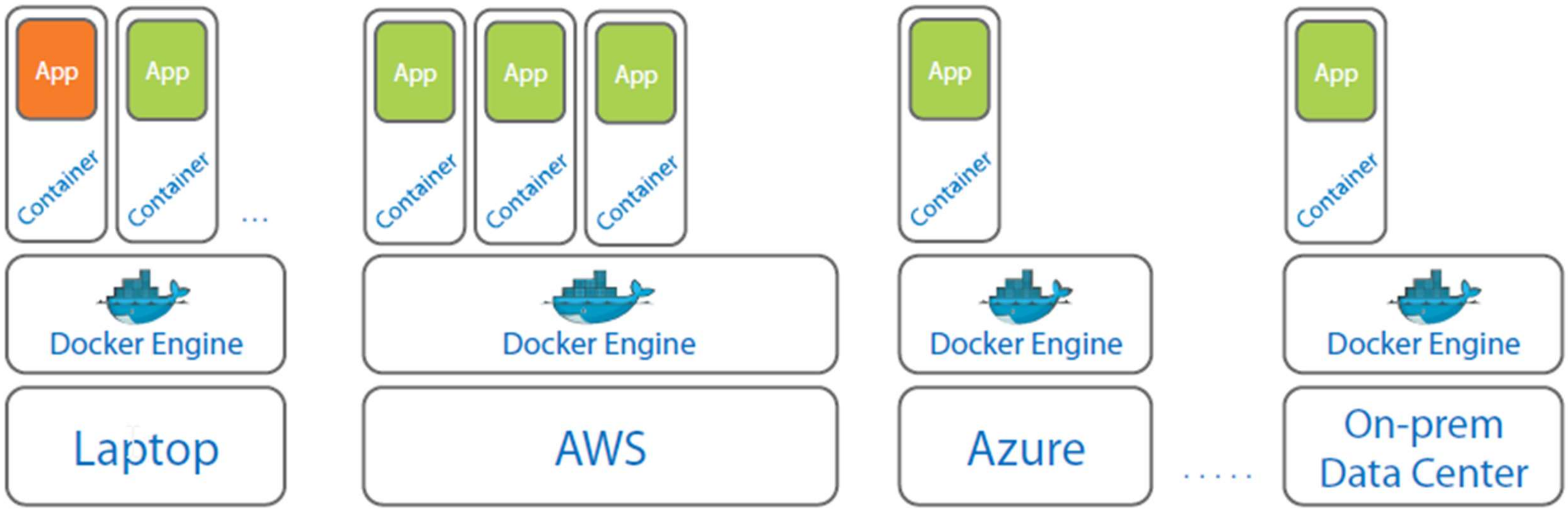
What is Docker?

- 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.

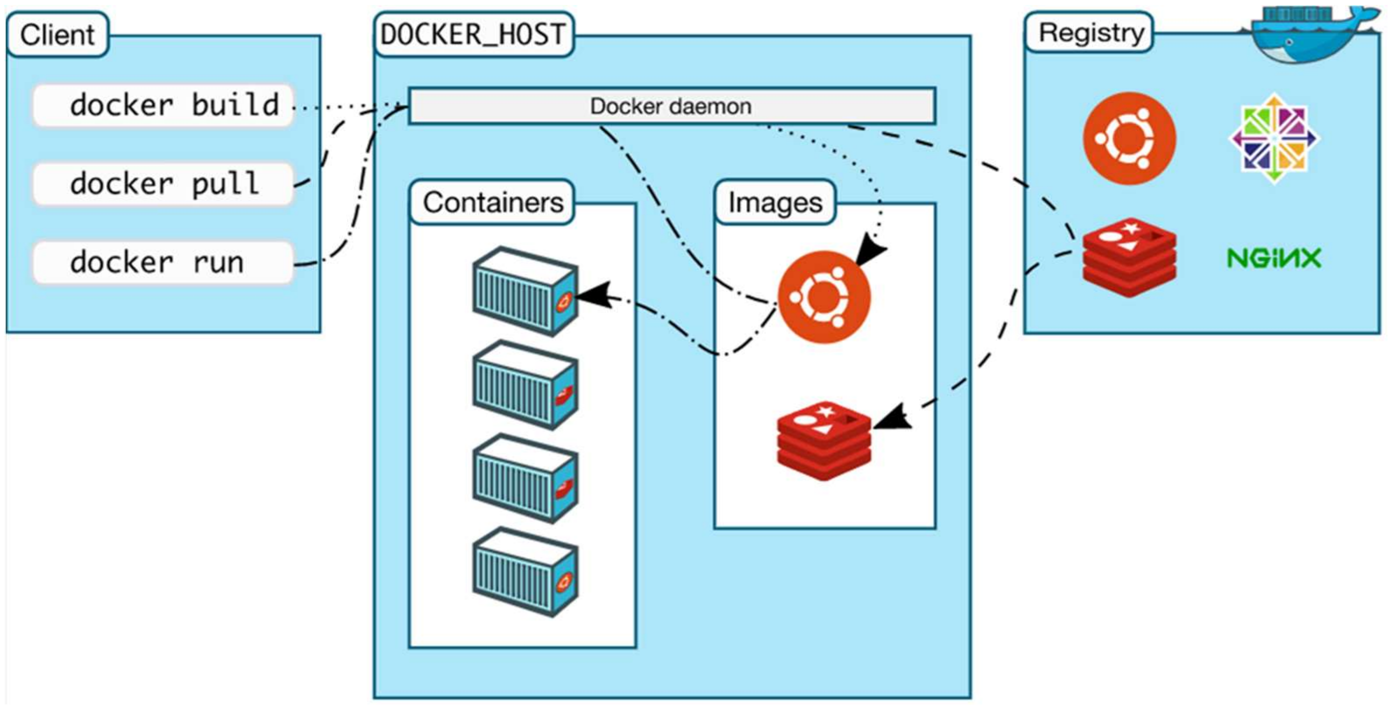
Docker Engine



Docker can run anywhere

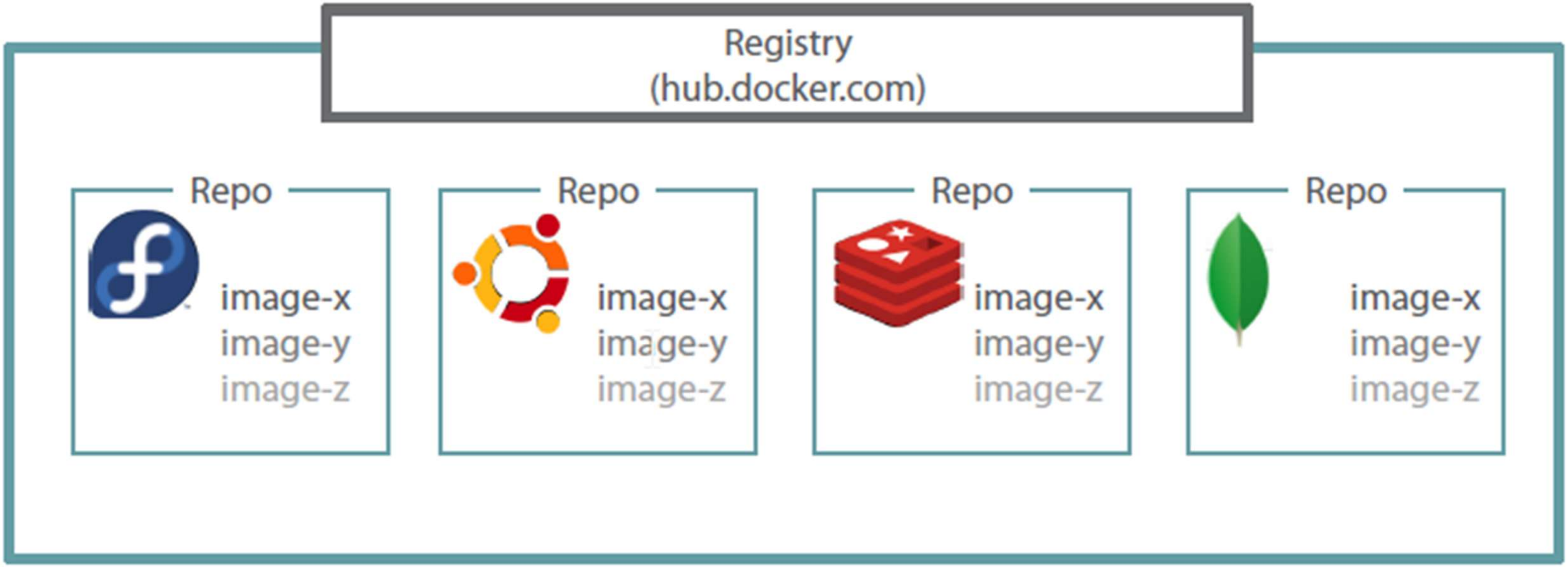


Docker Architecture



- Docker uses a client-server architecture.
- Docker client talks to the Docker daemon
- The Docker client and daemon can run on the same system, or can connect a client to a remote Docker daemon.
- The Docker client and daemon communicate using a REST API

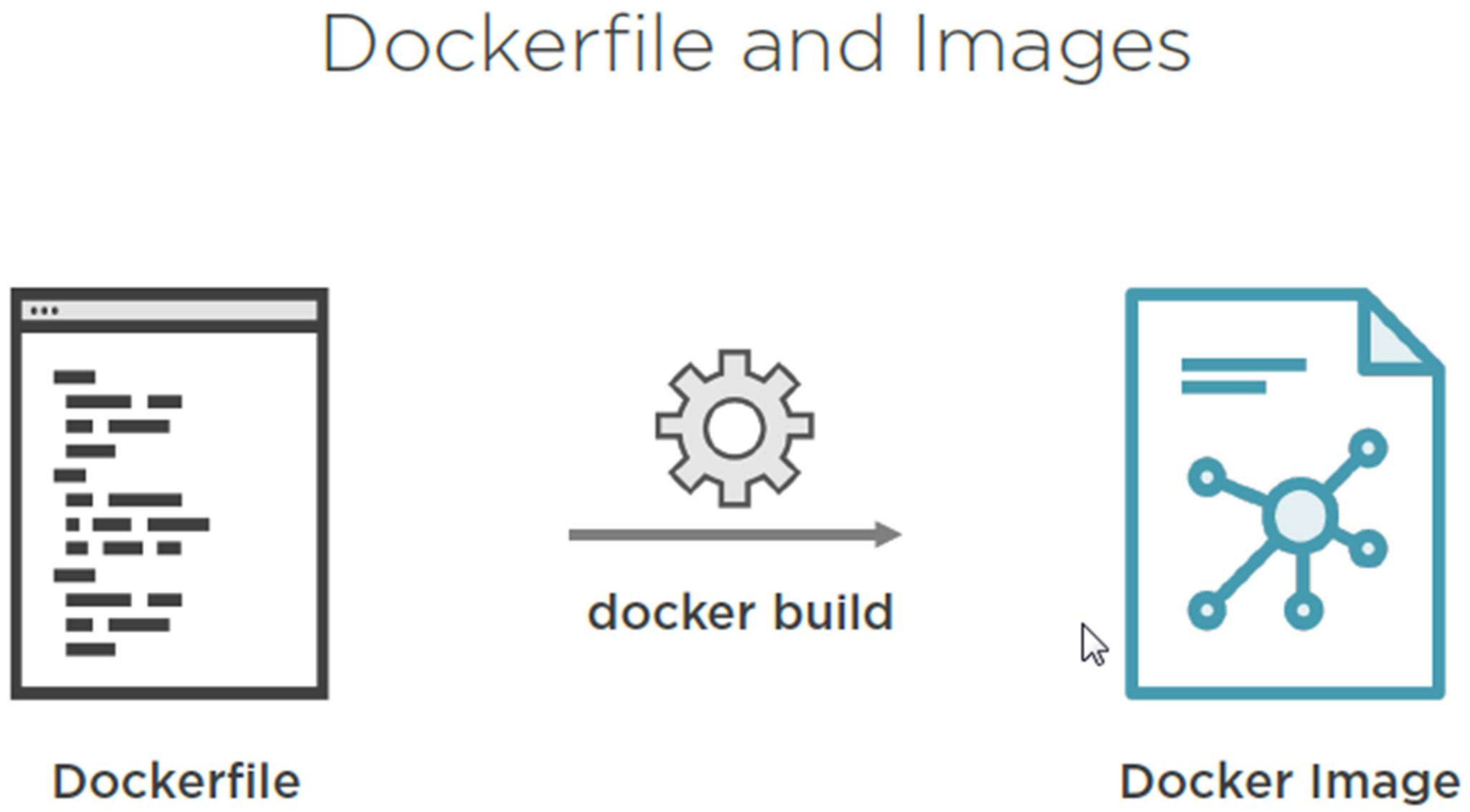
Docker Registry



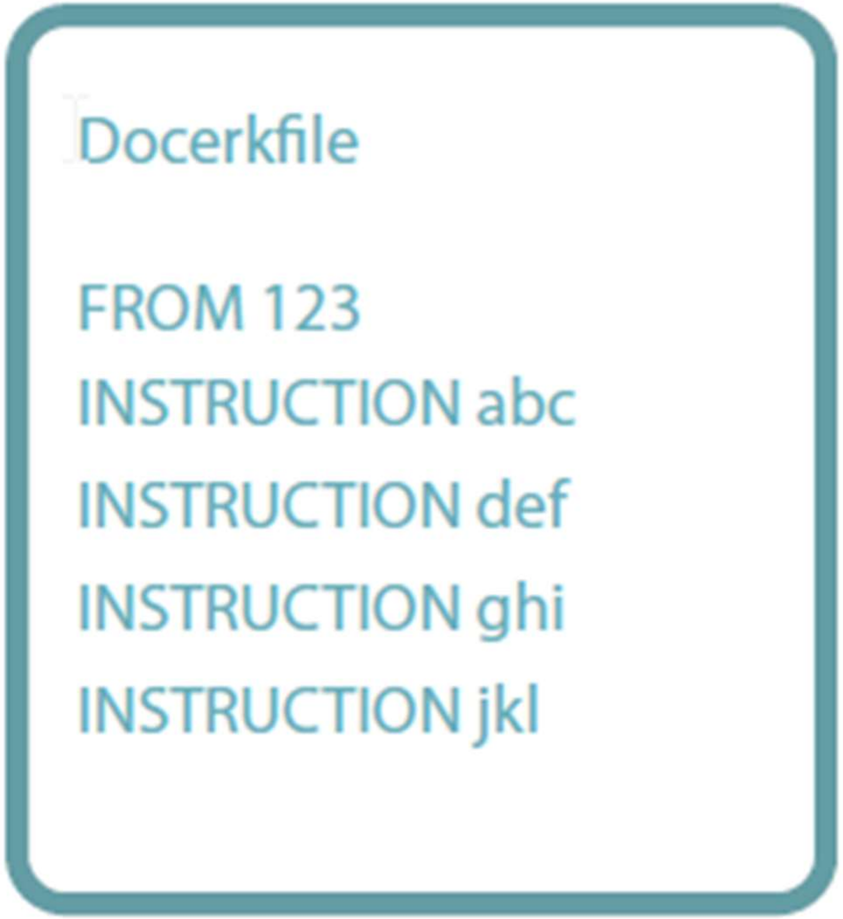
Hands-On

Container Images and Dockerfile

Dockerfile


Dockerfile and Images
docker build
Dockerfile
Docker Image

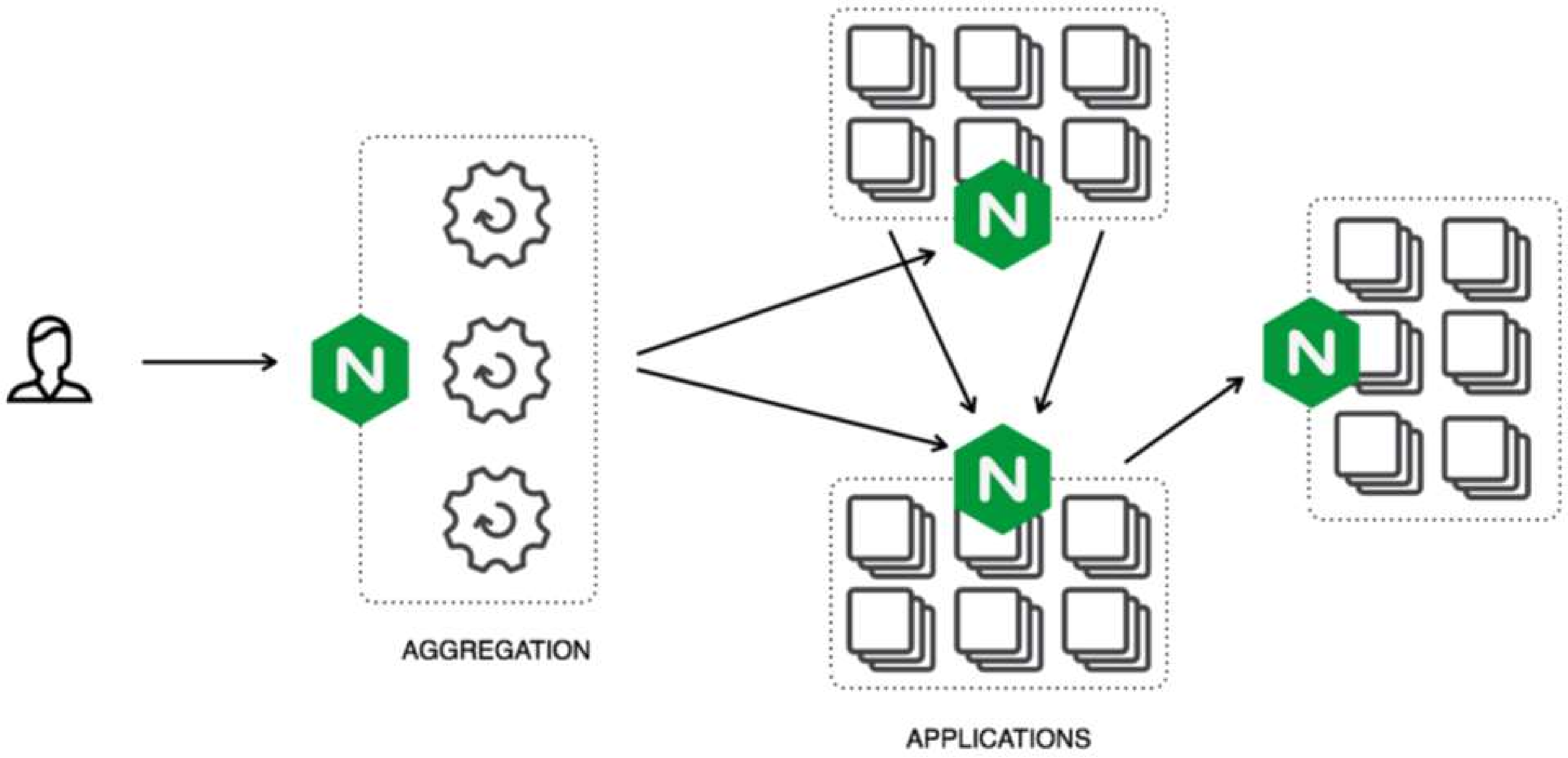
Dockerfile Template



```
Docerkfile

FROM 123
INSTRUCTION abc
INSTRUCTION def
INSTRUCTION ghi
INSTRUCTION jkl
```

Microservices



Thanks