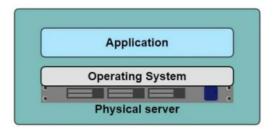
# One application on one physical server

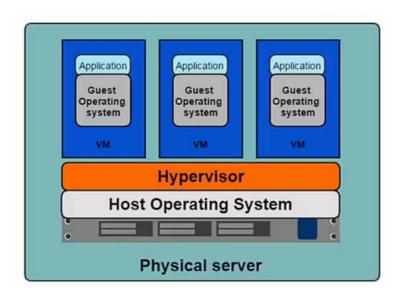


### application deployment problem in History

- Slow deployment
- Huge costs
- Wested resources
- Difficult to scale
- Vendor lock in

### Introduction to Virtualization

- One physical server can contain multiple applications
- Each application runs in a virtual machine



### **Limitation of VM**

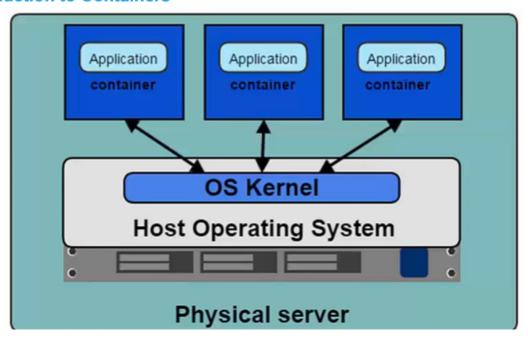
- Each VM still requires
  - CPU Allocation
  - Storage
  - o RAM
- The more VM the More resources

### **Introduction to Containers**

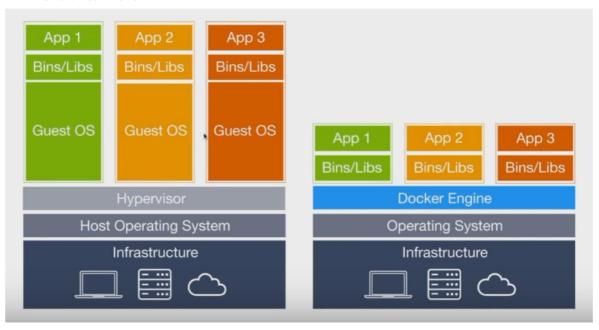
Container based virtualization uses the kernel on the host's operating system to run multiple guest instances

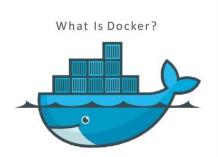
- Each guest instance is called a container
- Each container has its own
  - Root file system
  - Processes
  - Memory
  - Devices

### **Introduction to Containers**



### **VM Vs Containers**

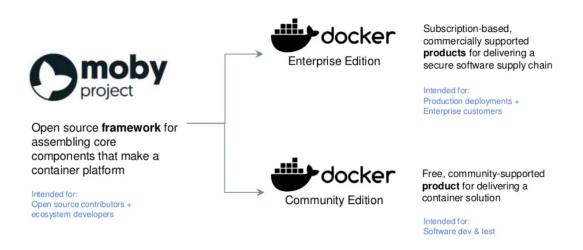




- · Lightweight, open, secure platform
- Simplify building, shipping, running apps
- Runs natively on Linux or Windows Server
- Runs on Windows or Mac Development machines (with a virtual machine)
- · Relies on "images" and "containers"



### The Docker Family Tree



### Container runtime engine

- Docker
- Cri-o
- PodDocker
- Cri-o
- Podman
- Rktlet

Microsoft Containers

### **Container Ecosystem**

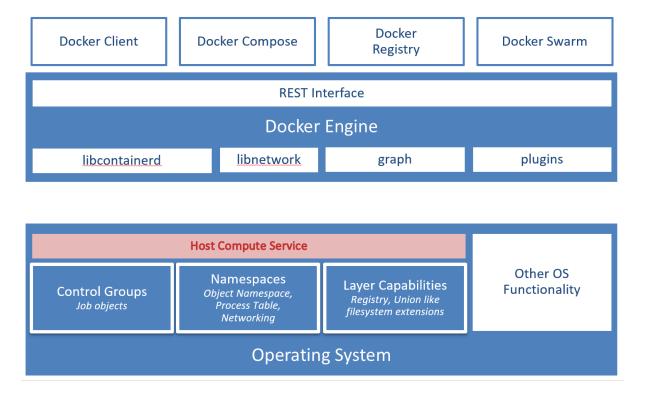
- Container
- Container image
- Container Runtime
- Container Registries

# Client Runtime Registry Daemon Pull Pull Remote API Containers Images Images Image registry

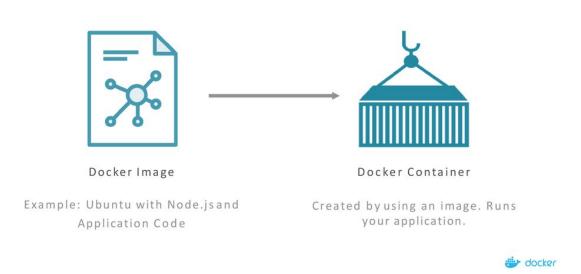
# **Architecture In Linux**

Docker Client	Docker Compose	Docker Registry	Docker Swarm
REST Interface			
Docker Engine			
libcontainerd	libnetwork	graph	plugins
<u>containerd</u> + <u>runc</u>			
Control Groups cgroups	Namespaces Pid, net, ipc, mnt, uts	Layer Capabilities Union Filesystems AUFS, btrfs, vfs, zfs*, DeviceMapper	Other OS Functionality
Operating System			

## **Architecture In Windows**



### The Role of Images and Containers



### Some Docker vocabulary



### **Docker Image**

The basis of a Docker container. Represents a full application



### **Docker Container**

The standard unit in which the application service resides and executes



### **Docker Engine**

Creates, ships and runs Docker containers deployable on a physical or virtual, host locally, in a datacenter or cloud service provider



### Registry Service (Docker Hub(Public) or Docker Trusted Registry(Private))

Cloud or server based storage and distribution service for your images docker

Sudo apt update
Sudo apt install docker.io
Sudo systemctl start docker
Sudo systemctl enable docker

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04