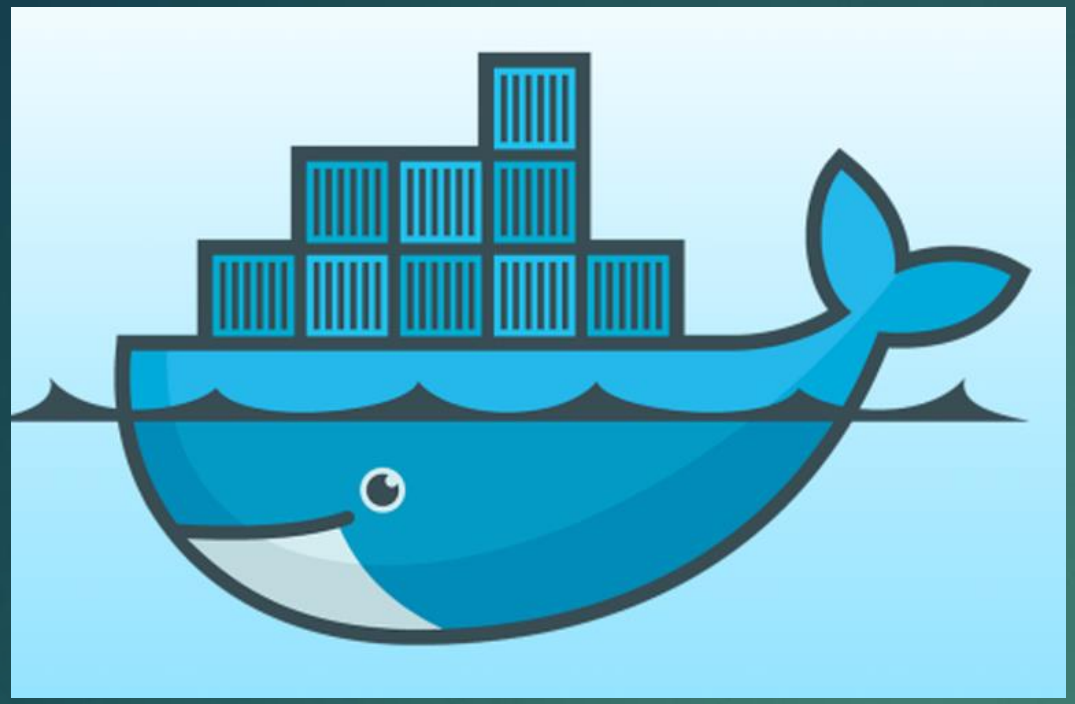


Deploy your dotnet core apps on Docker

MAHESH SABNIS – TRAINER & MCT & MICROSOFT MVP (VISUAL STUDIO AND TOOLS)

VIKRAM PENDSE – TECH MANAGER AT E-ZEST & MICROSOFT MVP (AZURE, WINDOWS PLATFORM)

Agenda

WHAT WE ARE GOING TO TALK

- DEPLOYMENT METHODOLOGIES
- VM
- NEED OF DOCKER CONTAINER
- DEMO

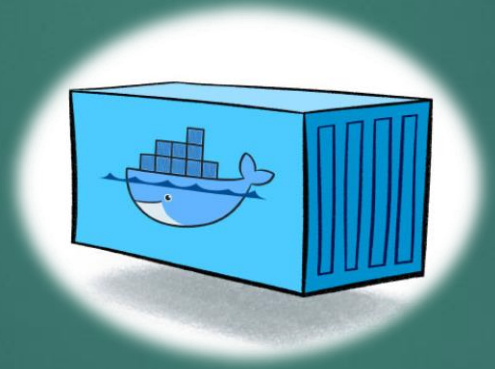
WHAT WE ARE **NOT** GOING TO TALK

- WINDOWS / WINDOWS HYPER-V CONTAINER
- INTER CONTAINER OPERATIONS / ORCHESTRATION
- KUBERNETICS

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.

[Source: en.wikipedia.org]

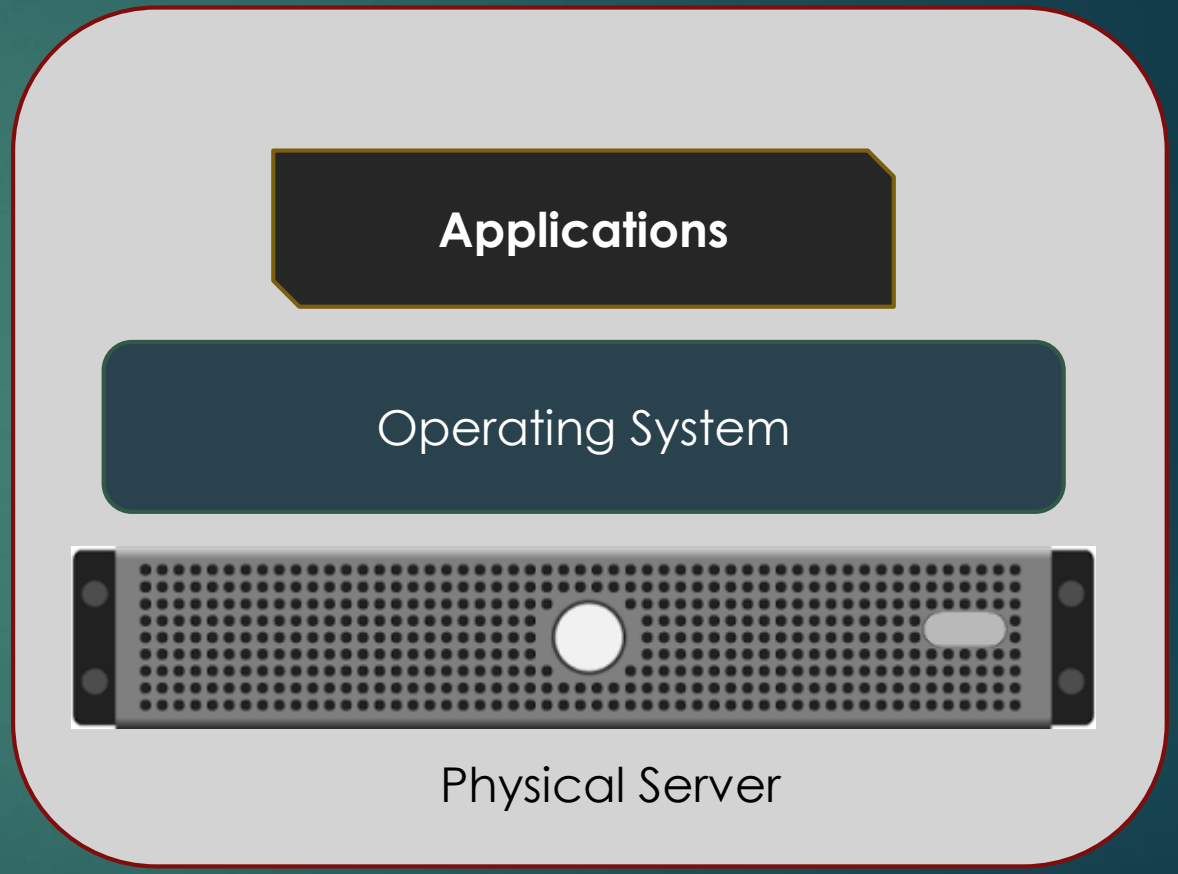


In short..

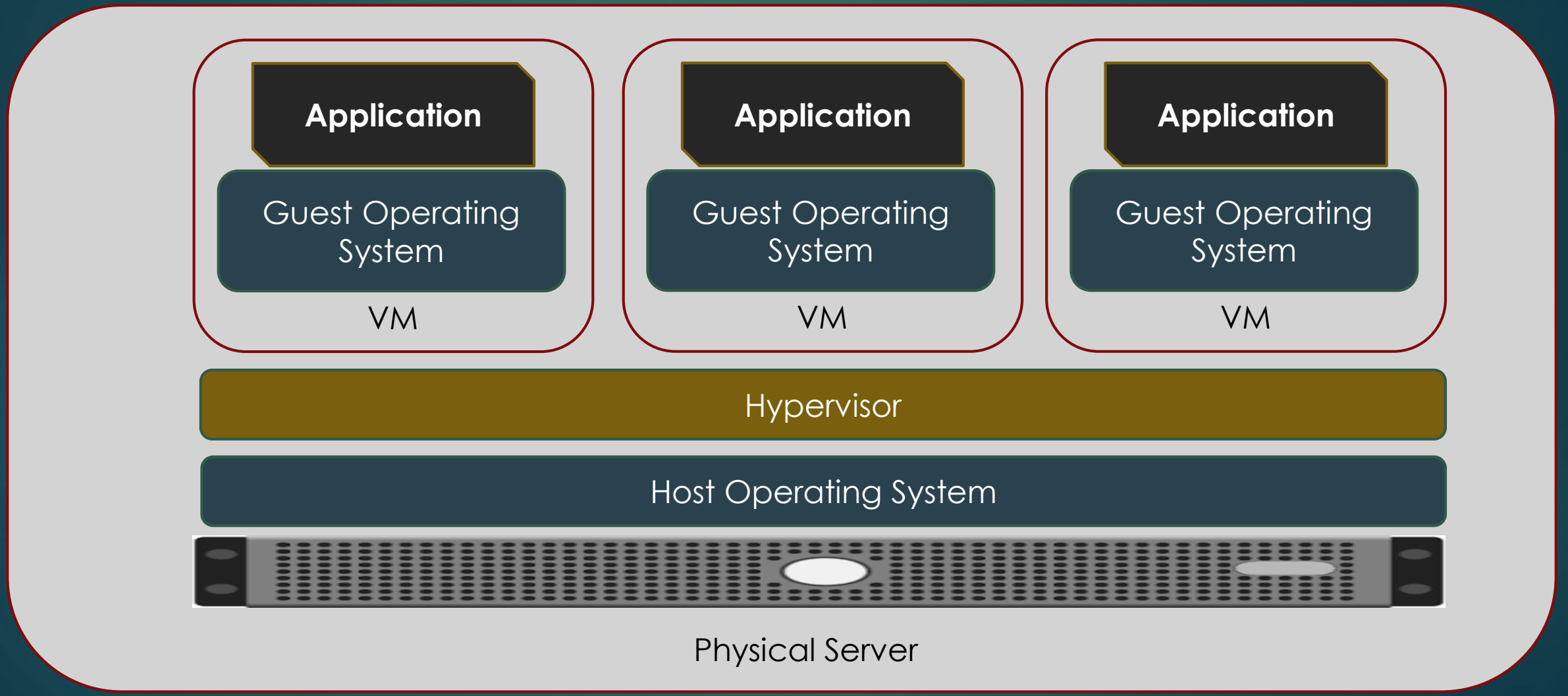
Docker is a platform for developing, shipping and running applications using container virtualization technology

Our haunted past

- Slow and sluggish deployments
- $ Cost $
- Waste of Resources
- Scaling issues
- Migrations issues
- Vendor dependency
  - IBM server to HP Server
  - Configuration dependencies and limitations



What we learn from our past



What we achieved using VMs?

- Resource Management
  - One Physical box divided/utilizes into multiple virtual instances/machines
- Scalability
- VM Mobility (Azure to On Premise and On Premise to Azure)
- On Demand (Azure !)
- Pay as you go (Azure !)

But…

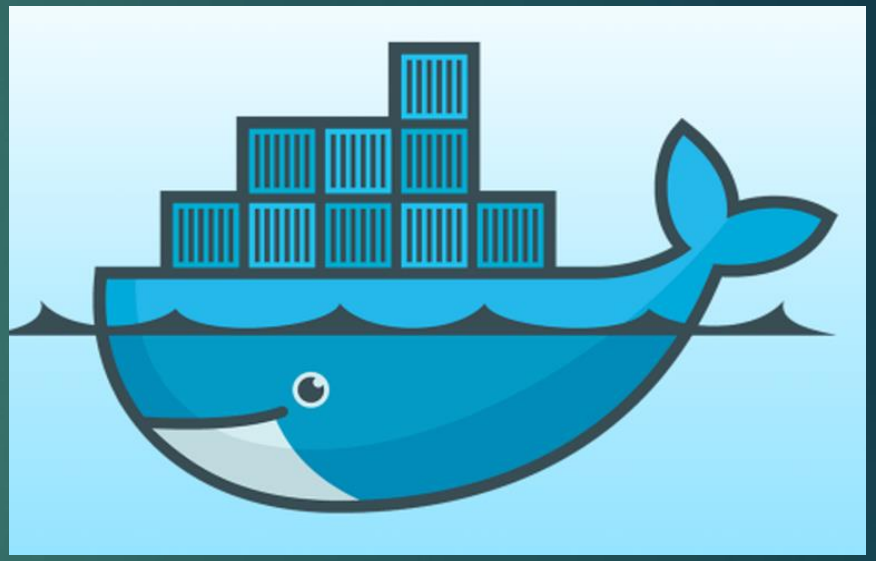
- VM requires
  - CPU
  - Storage
  - RAM
  - Full Guest OS in place ! ☹
- More number of VMs == More Resources consumption
- Most of the time Guest OS is Waste OS ! (Resource wastage)
- Pay for resources which you are not using while you go

Introduction to Containers

Container based virtualization uses the Kernel of Host OS to run multiple guest instances

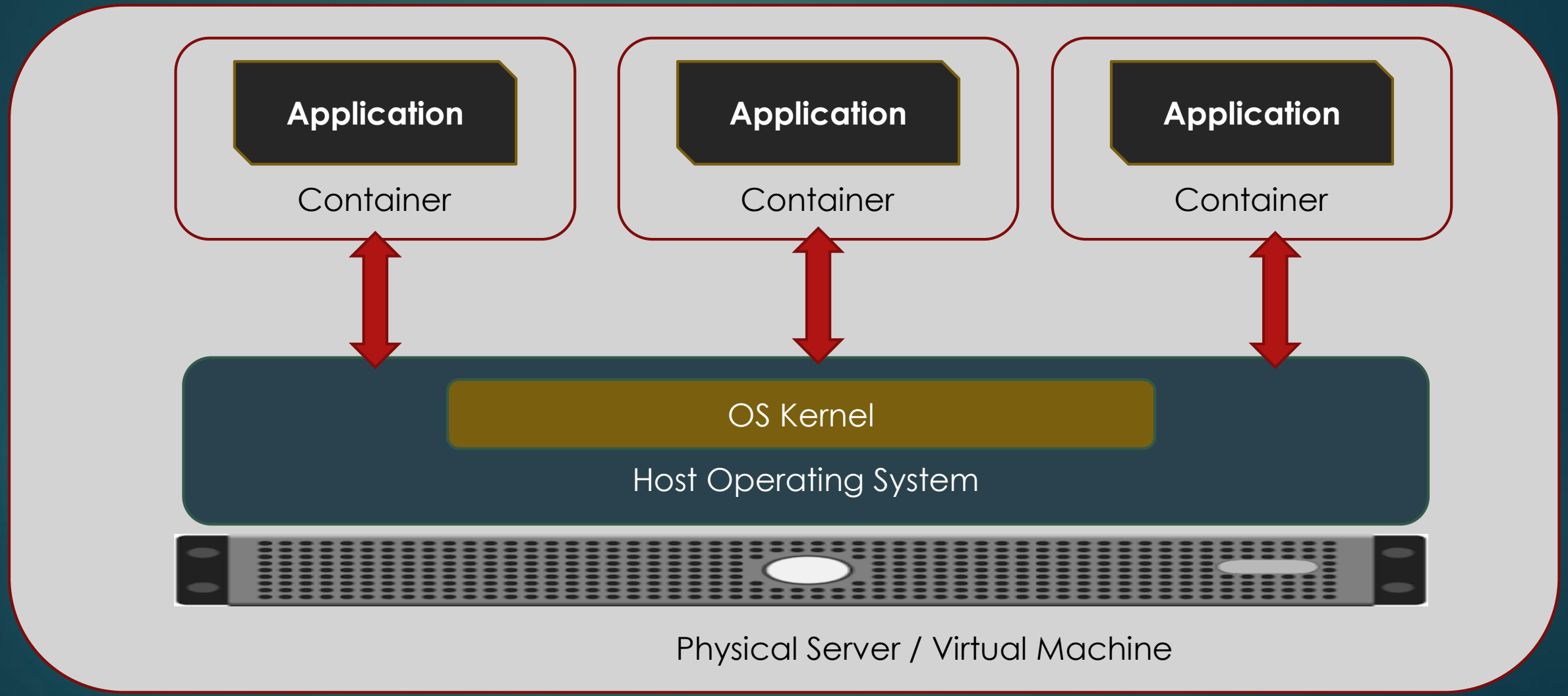
..and each guest instance is nothing but a "Container"

..it looks like a "VM" from outside but it is not !

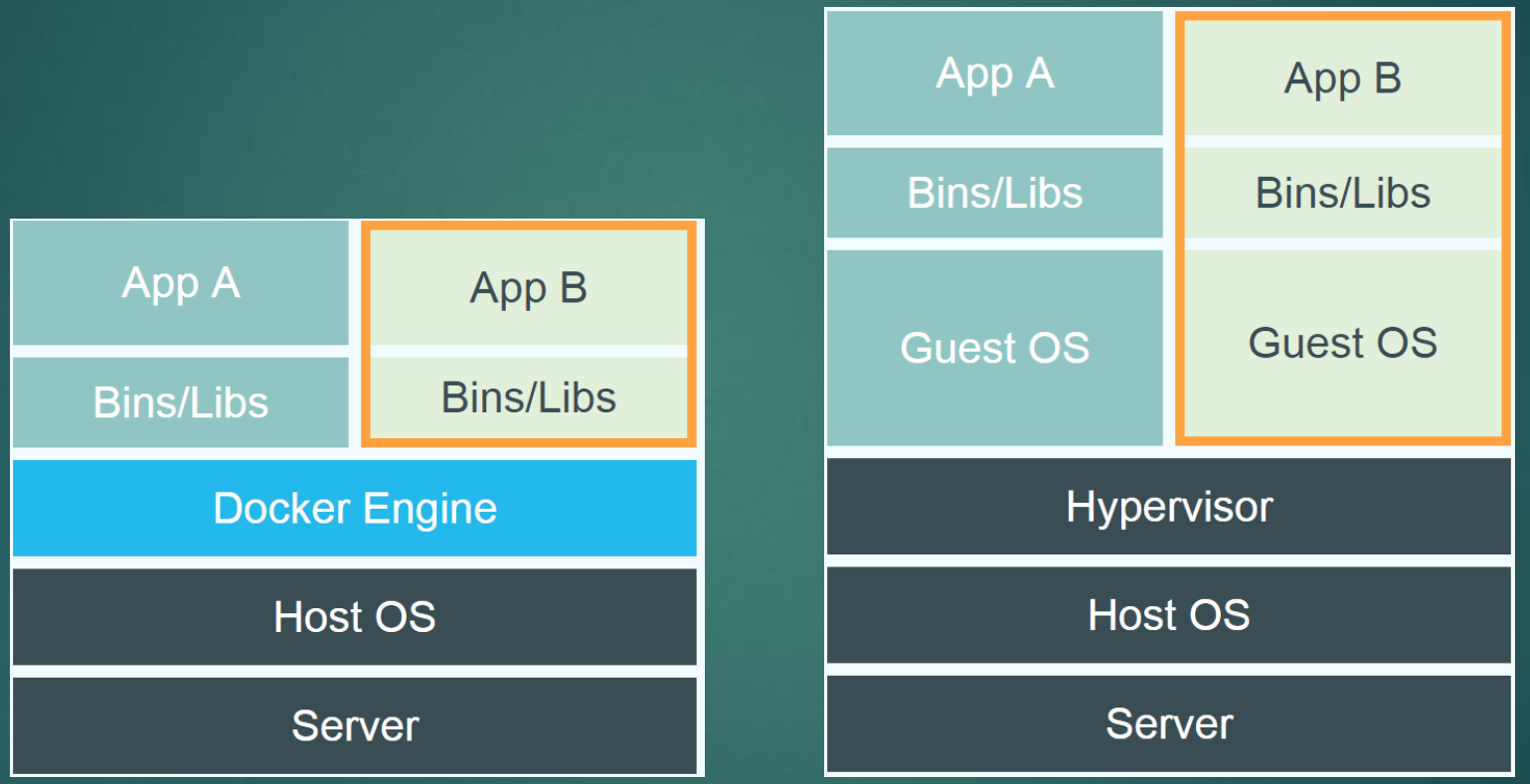


- Own File System / Root File System
- Processes
- Memory
- Devices
- Network Ports

Container


Application
Container
Application
Container
Application
Container
OS Kernel
Host Operating System
Physical Server / Virtual Machine

Docker Vs Virtual Machine



Why Docker?

- Microservices
- Virtual Machines
  - Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.
- Docker
  - The Docker Engine container comprises just the application and its dependencies.
  - It runs as an isolated process in user space on the host operating system, sharing the kernel with other containers.
- Docker provides base images that contain OS installations we can start from: The OS is not more than an application running on the Kernel...

Demo

Resources

- docker.com/Microsoft
- Channel 9 – Search for "Docker for .NET Developers" series
- DotNetCurry.com
  - **http://www.dotnetcurry.com/aspnet/1254/using-docker-container-aspnet-core-deployment**

Thanks !

- Mahesh Sabnis (@MaheshDotNet)
- Vikram Pendse (@VikramPendse) – vikram.pendse@e-zest.in
- PUG – facebook.com/puneusergroup or @PuneUserGroup (Twitter)
- www.puneusergroup.org
- PUG Events & Feedback :
  - Mahesh Mitkari (Mahesh@puneusergroup.org)
  - Vikram
  - Mayur

bit.ly/AzureQnA

MICROSOFT AZURE

INTERVIEW QUESTIONS

Your definitive guide to the best Microsoft Azure answers

Kunal Chandratre (Microsoft Azure MVP)
kunalchandratre@gmail.com

Vikram Pendse (Microsoft Azure and Windows Platform MVP)
vikrampendse@hotmail.com