# Windows/Hyper-V Container

Seung Joo Baek
Sr. Technical Evangelist
Microsoft Korea

# Microsoft and Docker

August 19, 2015

## INTRODUCING THE TECHNICAL PREVIEW OF DOCKER ENGINE FOR WINDOWS SERVER 2016

*written by Arnaud Porterie, Docker Senior Engineering Manager*

It's here…


As core engineer on the Docker Engine team, I naturally spend most of time in Linux. Recently that has been changing: in April, we released a Windows version of the Docker client. Through this process, we have been working closely with Microsoft developers and showing progress along the way, like what was demonstrated at the Build conference and DockerCon 2015.

One question I get a lot besides "When are you merging my PR?" is "When will Docker run on Windows?" The first question requires a blog post of it's own… but the second question now has a rather exciting answer.

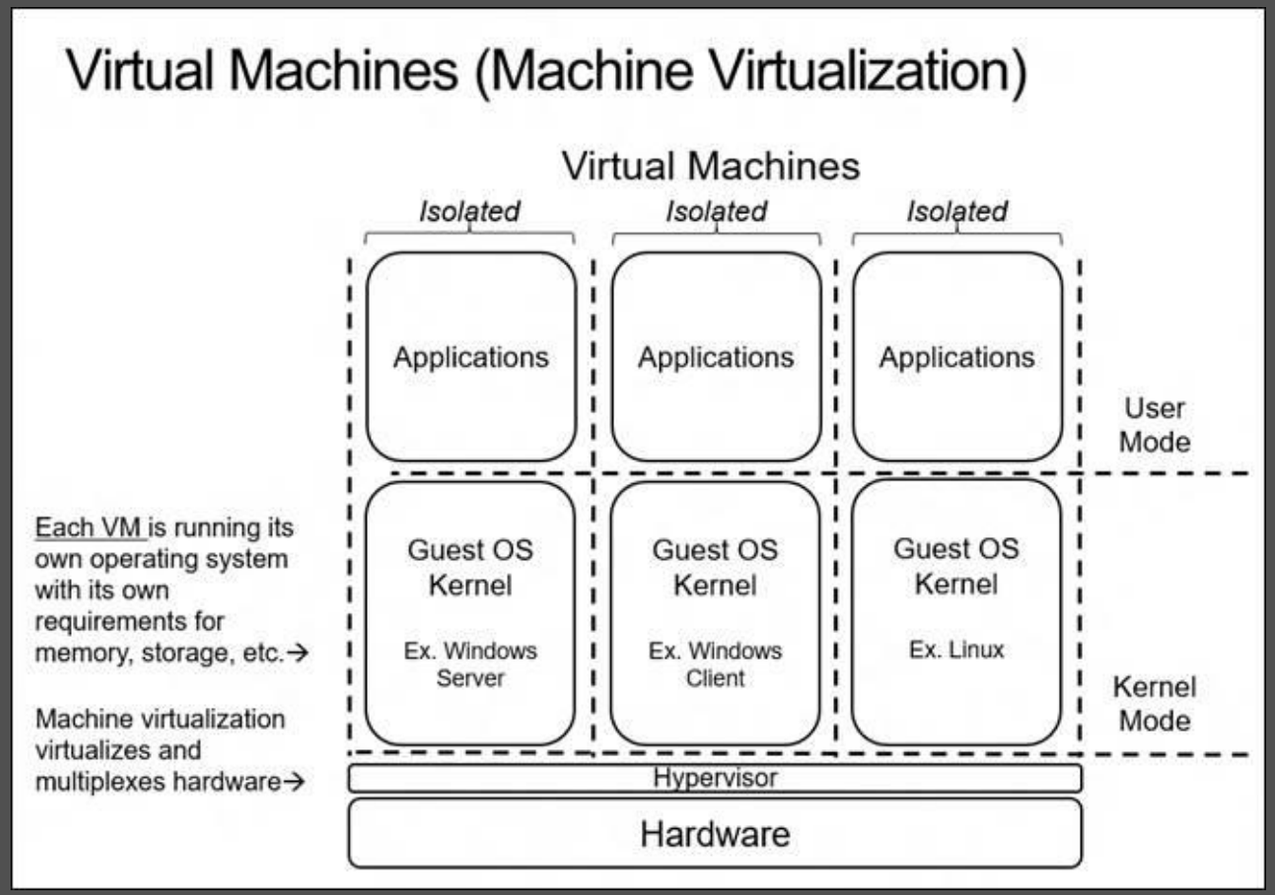
This week marks a huge leap forward as we release a tech preview of Docker

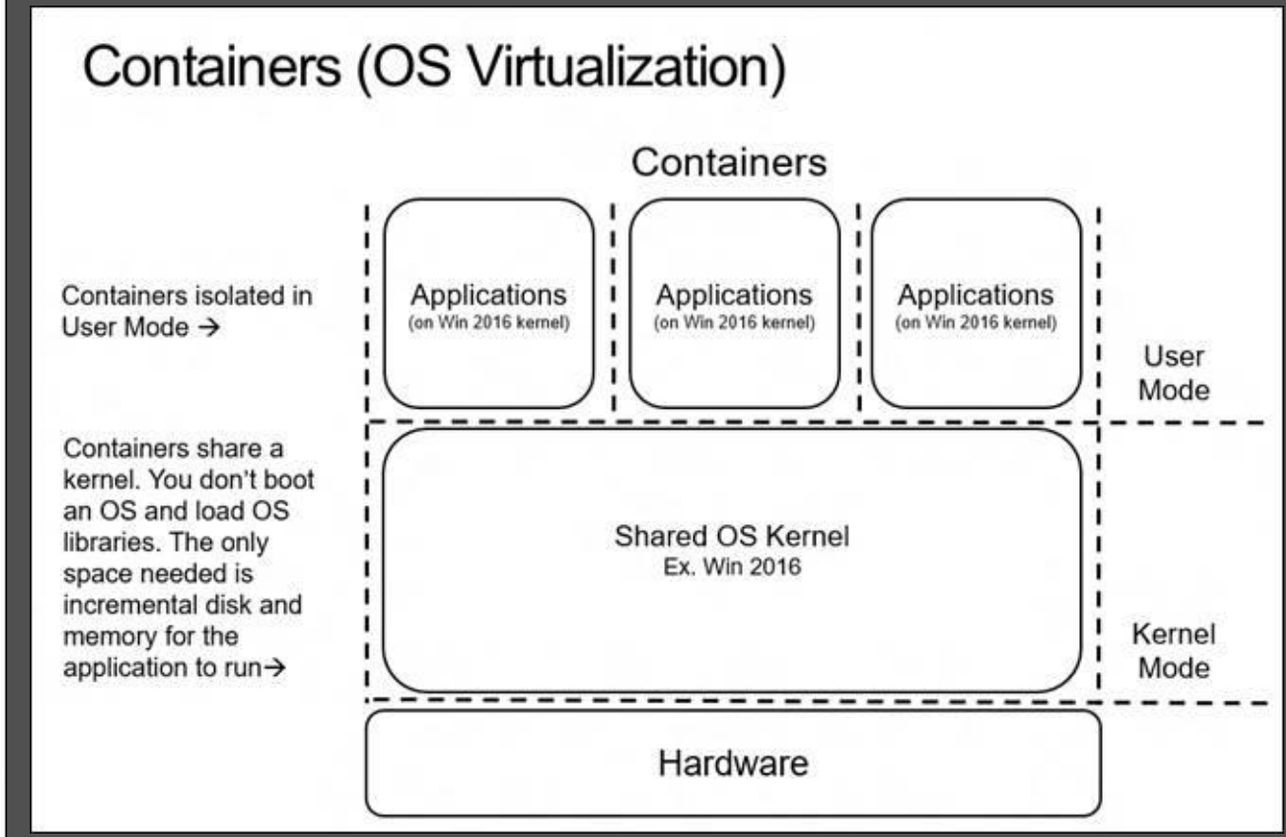
## Docker daemon for Windows Server in a nutshell

Many are confused by what it means to run Docker on Windows Server. Here are a few key points to help clarify the process and technical details:

- Docker Engine for Windows Server port is not a fork, nor a different project: it's the same open source code base being built for Linux and Windows
- The Technical Preview of Docker for Windows Server isn't feature complete yet (and taking into account that the feature sets will never be exactly identical given the differences of the underlying platforms), most of the Docker commands you already know will work as expected on Windows Server. For example, you can write a `Dockerfile` and `docker build` as you would on Linux.
- The Docker daemon for Windows Server doesn't run Linux images! No virtualization is involved. The Windows Server Containers reuse the host kernel and create a sandboxed environment for the process, exactly like it does on Linux.

# Comparison


Virtual Machines (Machine Virtualization)
Virtual Machines
Isolated
Isolated
Isolated
Applications
Applications
Applications
User Mode
Guest OS Kernel
Ex. Windows Server
Guest OS Kernel
Ex. Windows Client
Guest OS Kernel
Ex. Linux
Kernel Mode
Each VM is running its own operating system with its own requirements for memory, storage, etc.→
Machine virtualization virtualizes and multiplexes hardware→
Hypervisor
Hardware


Containers (OS Virtualization)
Containers
Containers isolated in User Mode →
Applications
(on Win 2016 kernel)
Applications
(on Win 2016 kernel)
Applications
(on Win 2016 kernel)
User Mode
Containers share a kernel. You don't boot an OS and load OS libraries. The only space needed is incremental disk and memory for the application to run→
Shared OS Kernel
Ex. Win 2016
Kernel Mode
Hardware

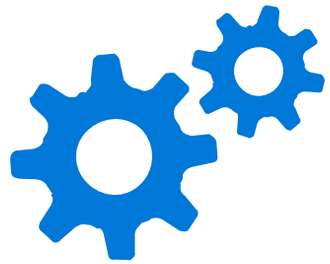
# Container Operating System Environments

## Nano Server

Highly Optimized

Born in the cloud applications

## Server Core

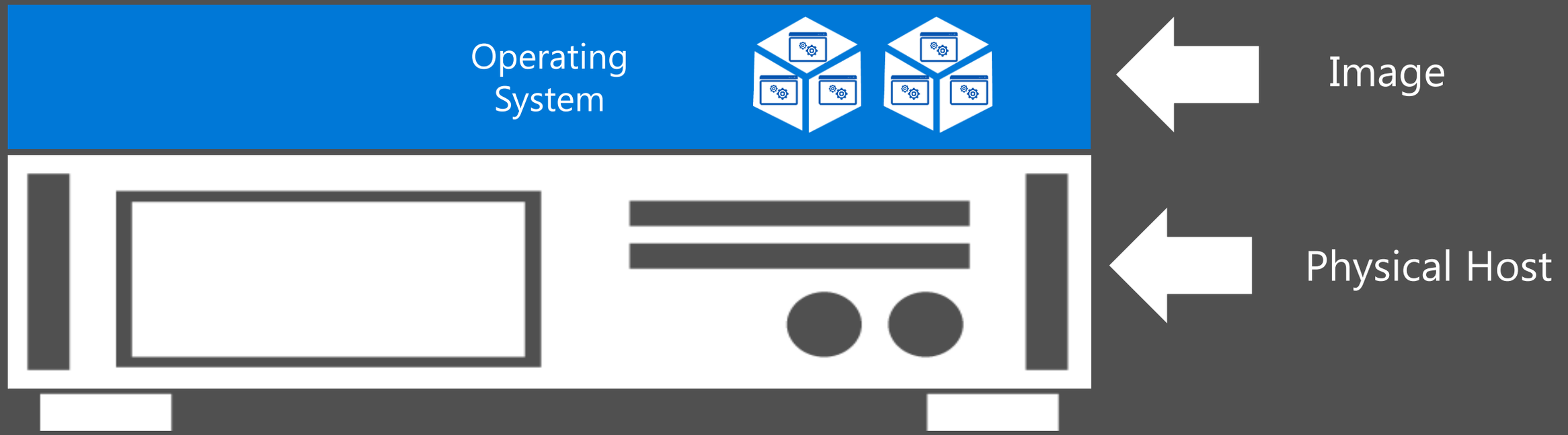


Highly Compatible

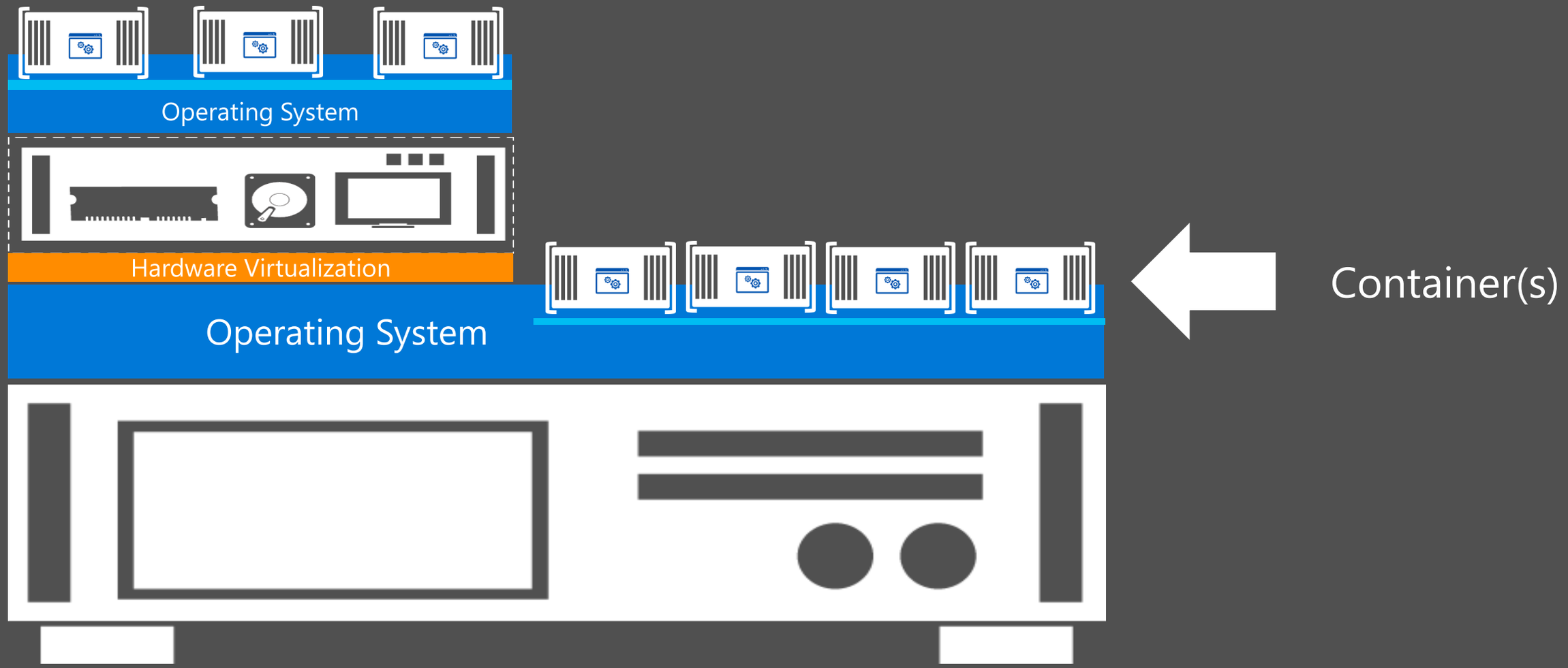
Traditional Applications

It’s not supported Windows Server GUI

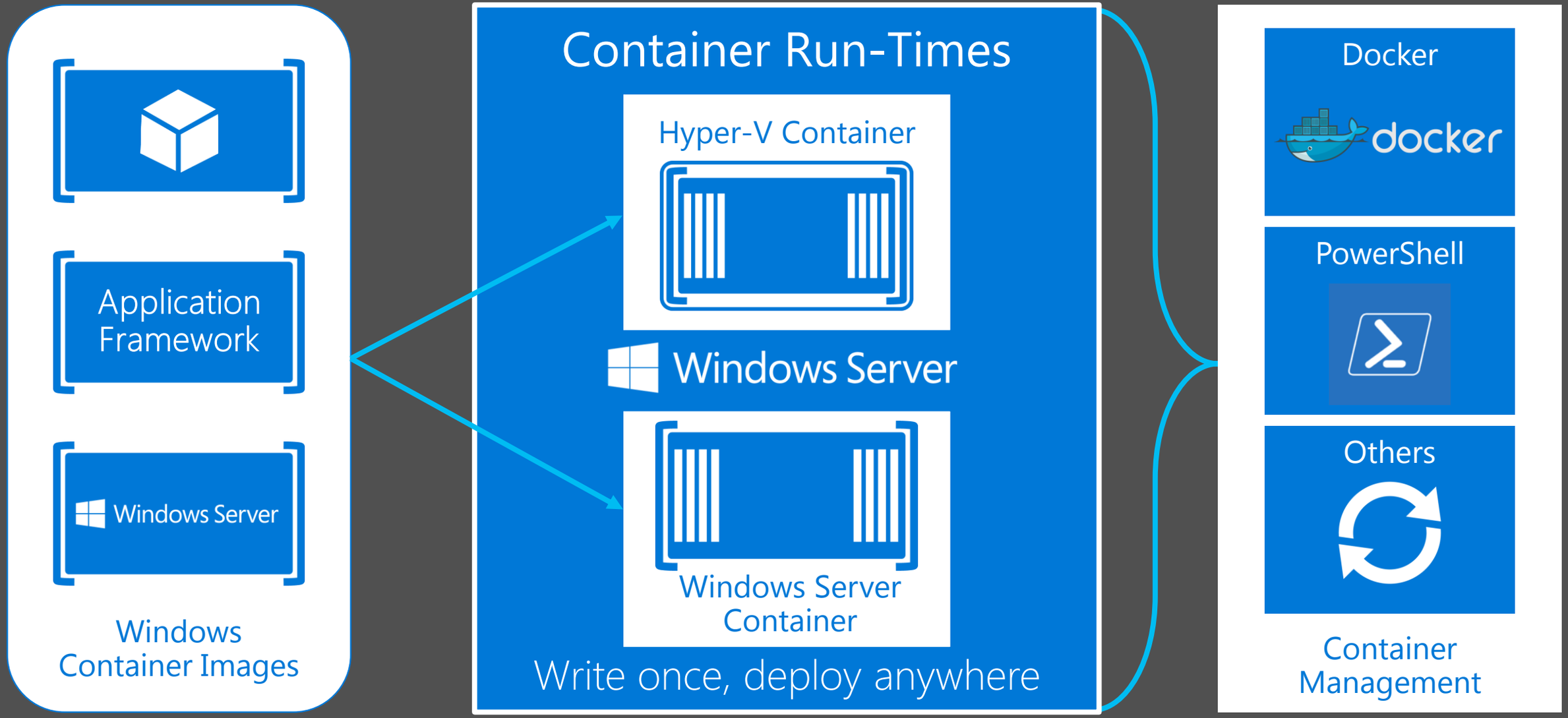
# Container Host



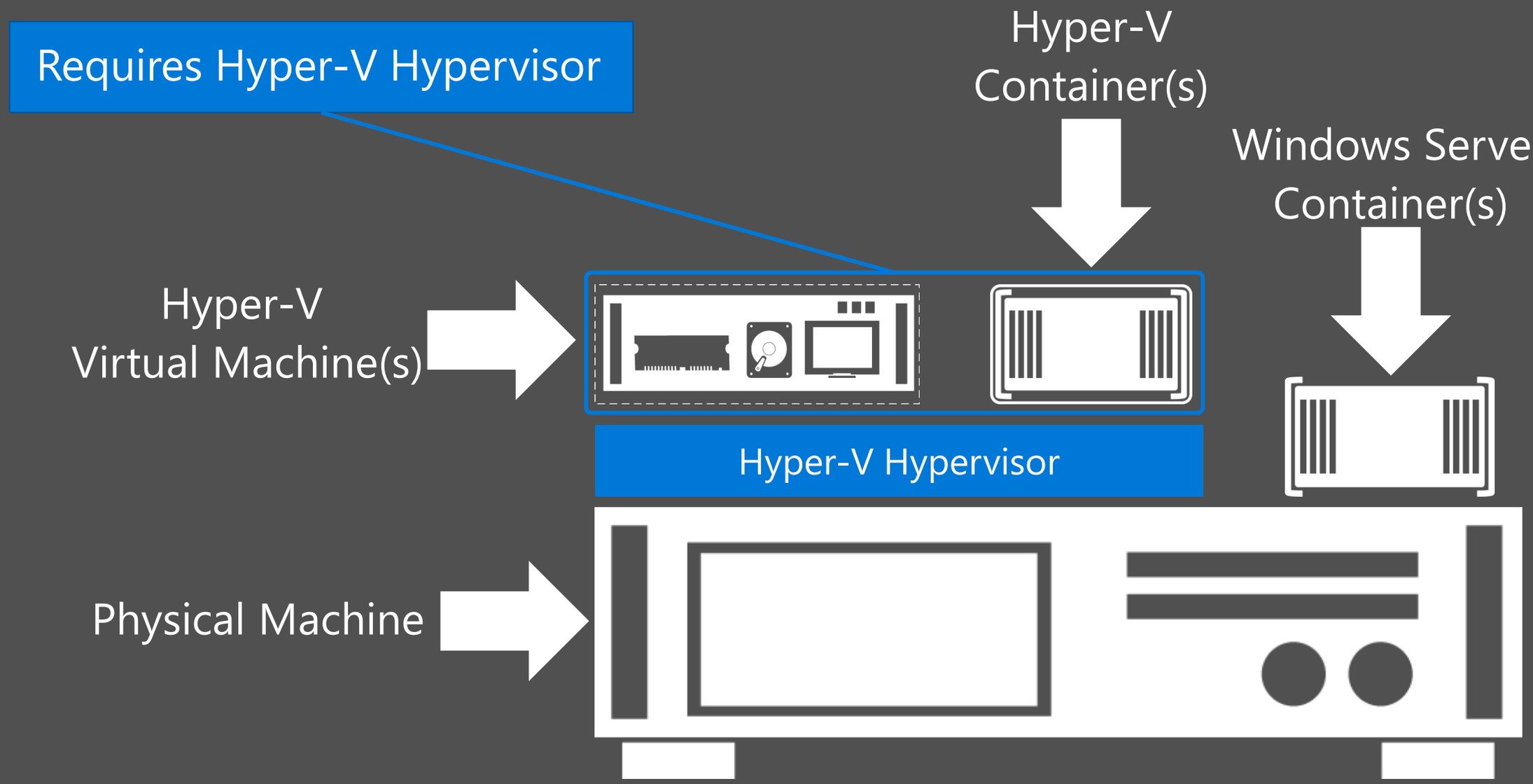
# Container Host



# Modern App Development, Flexible Isolation



# Deploying Containers



# *The right tools for you.*

## Development Environments

- Visual Studio
- eclipse
- Others…

## Container Management

- PowerShell
- Docker
- Others

## Container Technologies

- Windows Server
- Linux

## Microsoft Cloud

- Azure
- On Premises
- Service Provider

Microsoft