SYNERGETICS GET IT RIGHT | Microsoft Partner Gold Cloud Platform Silver Learning

Introduction to Containers

Module 1

Objective

- Need for Application Isolation
- Need for Portable Applications
- Disadvantages of Using Virtualization
- Introduction to Containers
- Kernel features for containers
- Containers on Windows and Linux platform

Need for Application Isolation

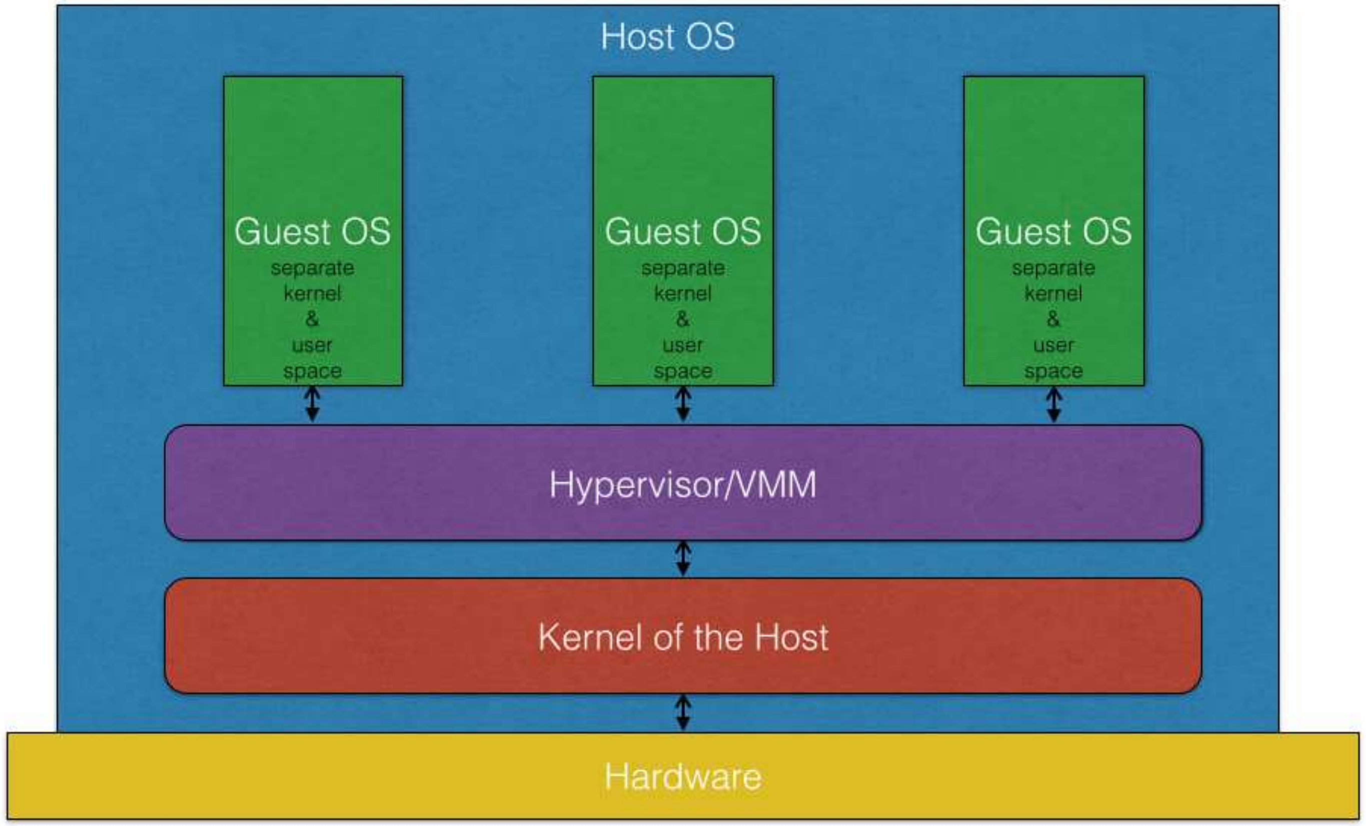
- Every Application has certain dependencies.
  - Libraries provide by Operating System
  - Third Party Libraries
- Change in Dependencies affects Application.
- Application should have its own sandbox.

Need for Portable applications

- Application goes through following environments:
  - Development
  - Testing
  - Staging
  - Production
- Managing Dependencies across all environments could be difficult.
- Creating a compatible dev-test environment may take considerable time.

Virtualization

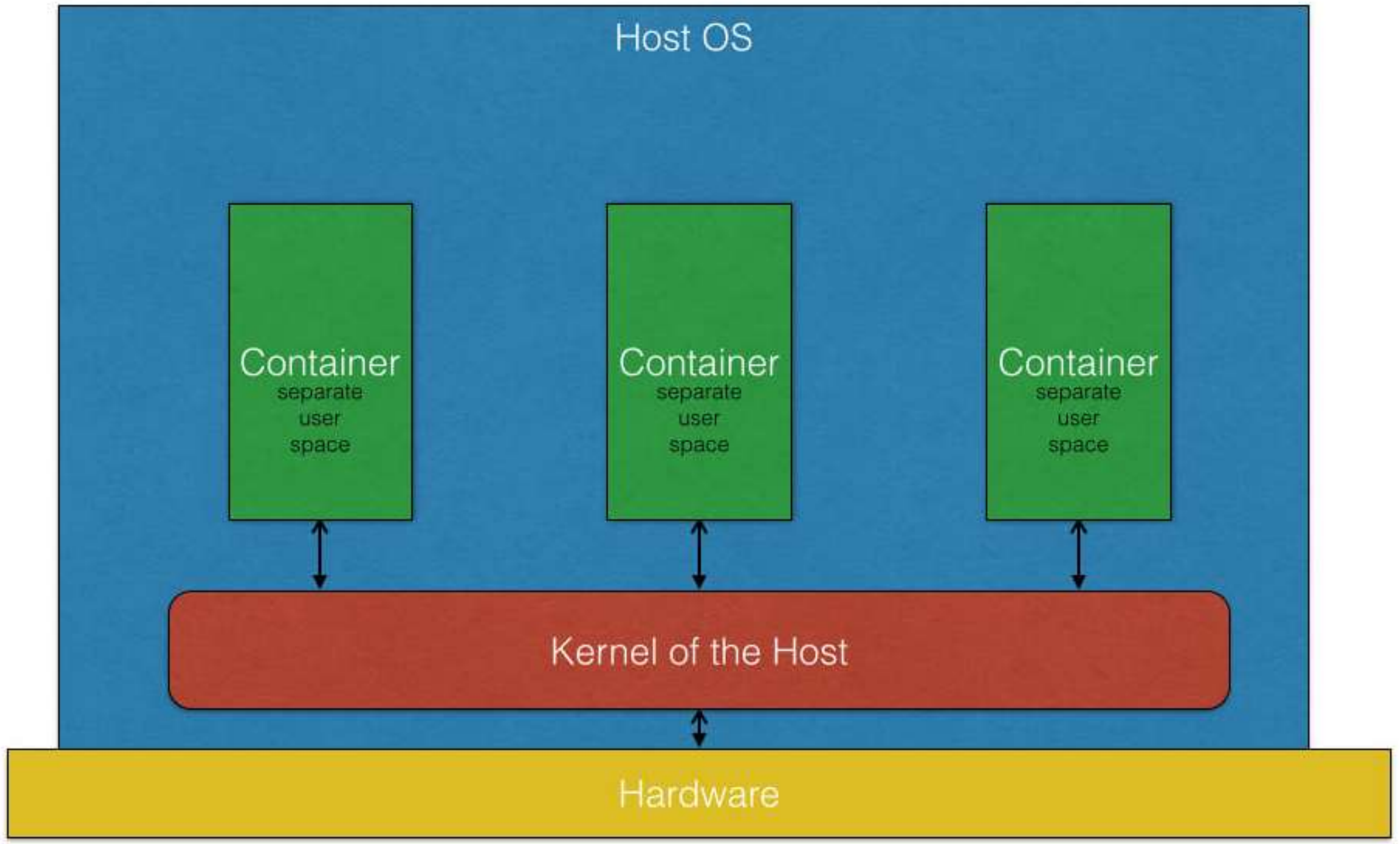
Complete Isolation [Virtualize Hardware and Operating System]
Time Consuming
Not ideal for isolating Individual application.


Host OS
Guest OS
separate kernel & user space
Guest OS
separate kernel & user space
Guest OS
separate kernel & user space
Hypervisor/VMM
Kernel of the Host
Hardware

Hypervisor based Virtualization

Introduction to Containers

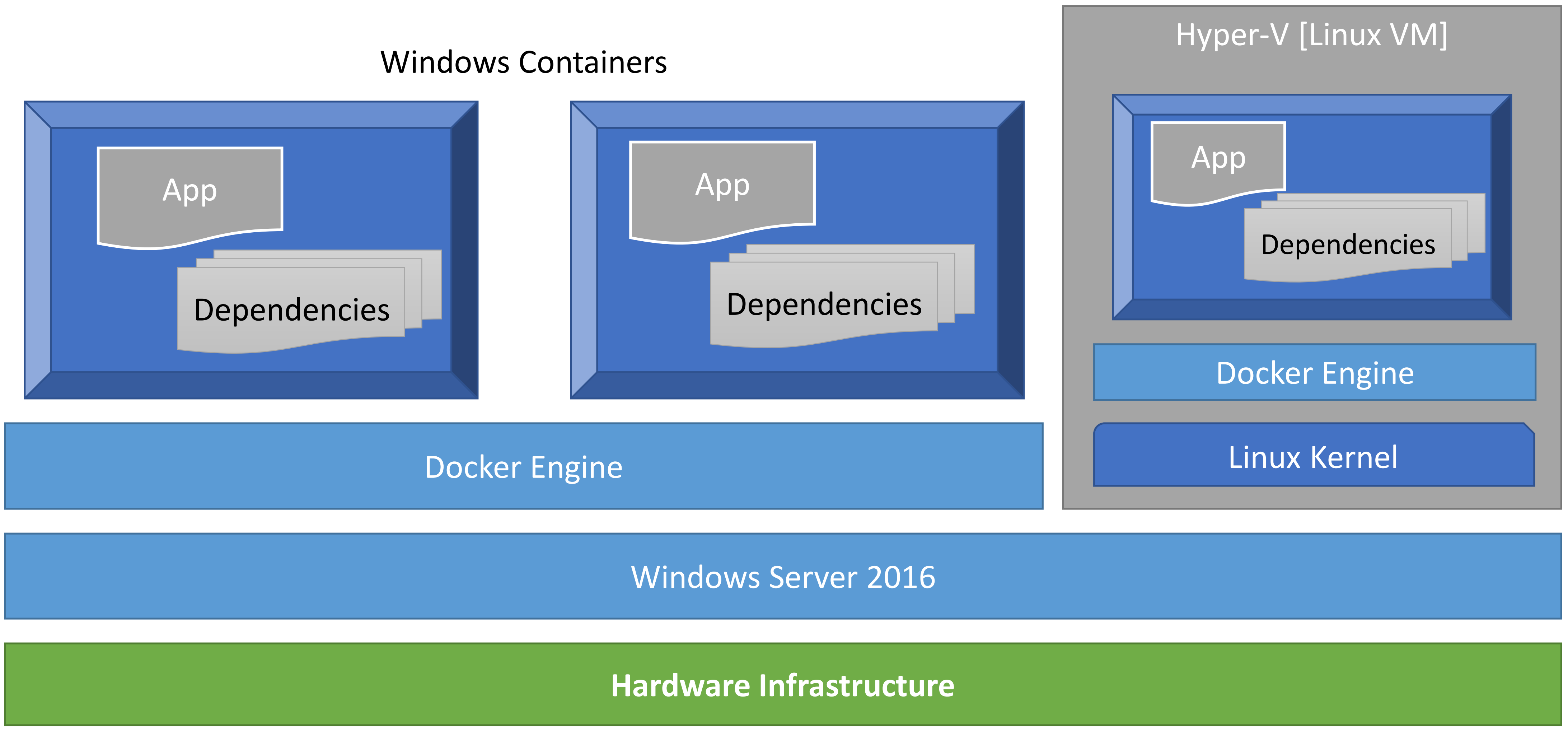
No Hardware Virtualization
Targeting One Application
Packs ALL dependencies of Target App
Execute in separate User-Space



Kernel Features for containers

- Operating System Kernel should provide following:
  - Control Groups [cgroups]: Resource Metering & limiting
  - Namespaces: provide processes with their own view of the system.

Containers for Windows and Linux

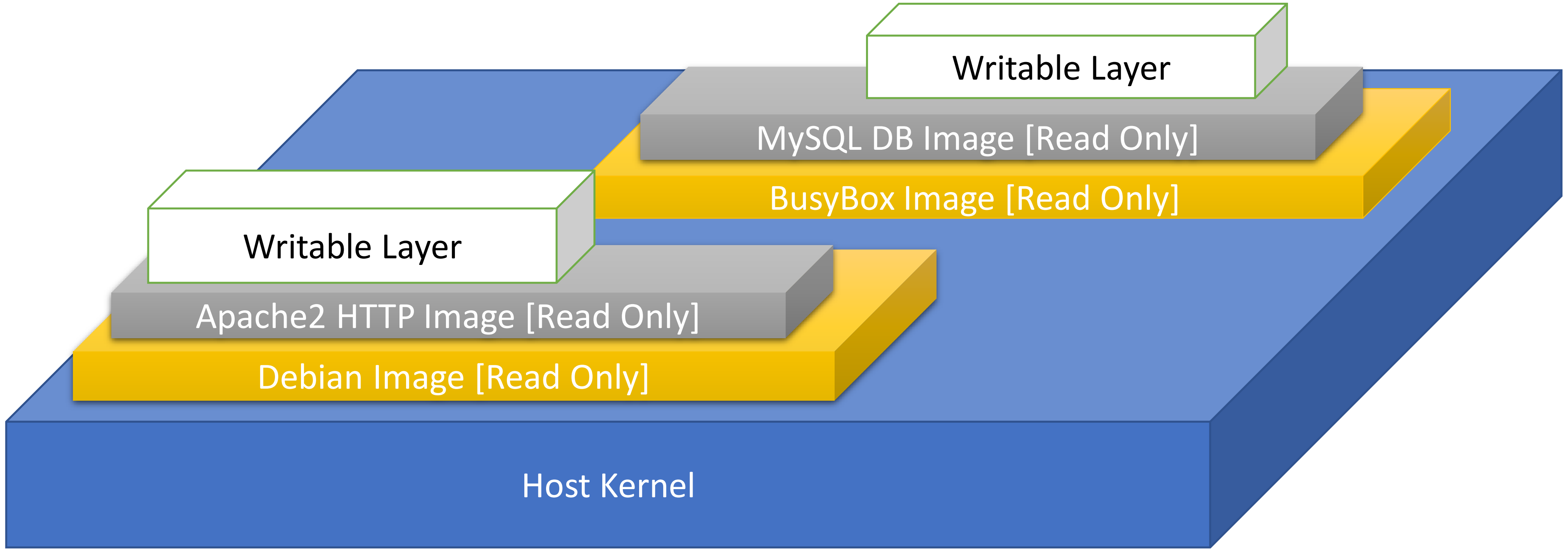


SYNERGETICS GET IT RIGHT | Microsoft Partner Gold Cloud Platform Silver Learning

Containers in details

Module 2

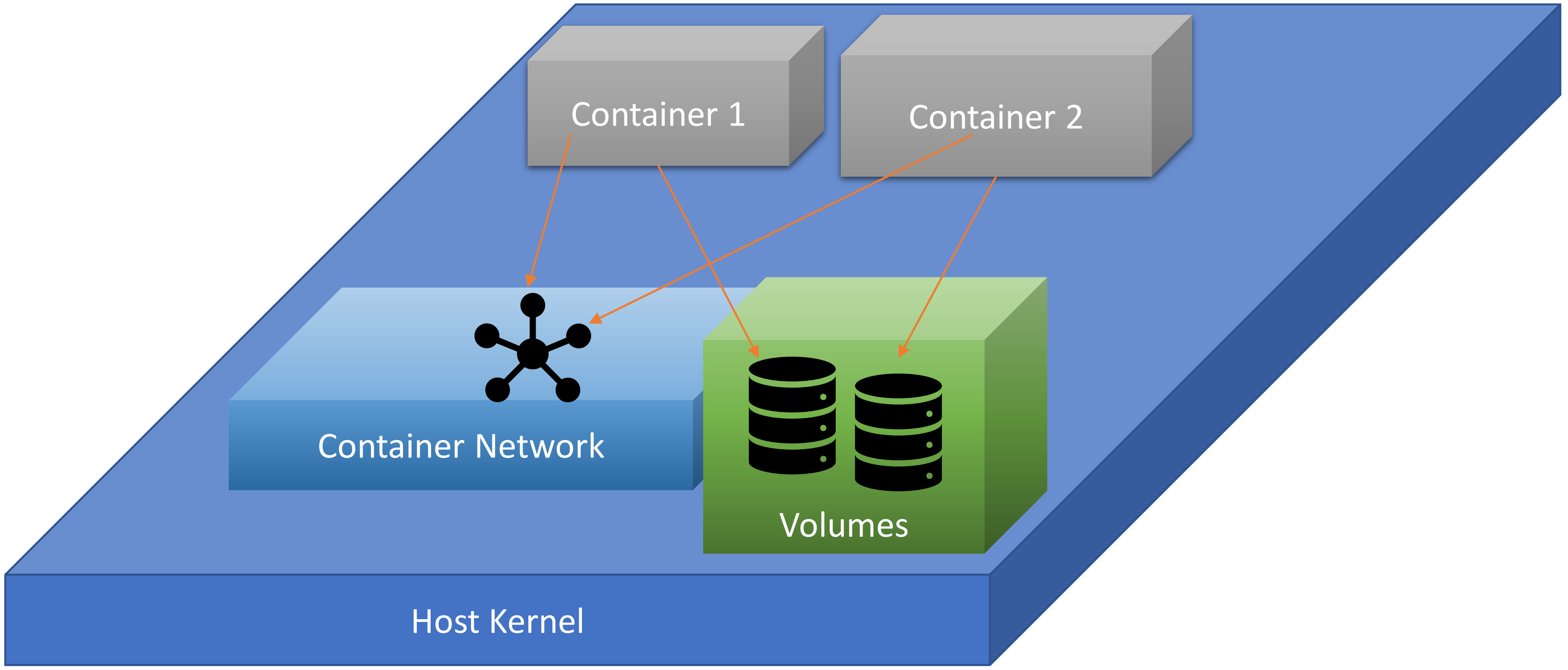
Container architecture


Writable Layer
MySQL DB Image [Read Only]
BusyBox Image [Read Only]
Writable Layer
Apache2 HTTP Image [Read Only]
Debian Image [Read Only]
Host Kernel

Container Architecture

- Container is a running instance of an Image.
- Made of lots of layers.
- Each layer is an Image. The topmost is Writable.
- The bottom most image is called Base Image.

Container Architecture



Inter container communication

- Factors affecting communication between TWO containers on same Host:
  - Does Network topology allows to connect containers NIC?
  - Does Firewall allows particular connection?
- Factors affecting containers communication to outside host
  - Is Host system forwarding its IP packets.
  - Firewall allows this particular connection.
- We will discuss more about it, in Docker architecture

Running Containers

- Containers can run in following modes:
  - As Daemon

    Containers starts and continue execution in background. Most common for production environment.

    Examples : WebApp in container.
  - As Interactive

    Containers start with interactive shell [eg Bash in Linux]. Allows host user to write commands and get immediate results.

    Examples: AzureCLI in container.

SYNERGETICS GET IT RIGHT | Microsoft Partner Gold Cloud Platform Silver Learning

Introduction to Docker

Module 3

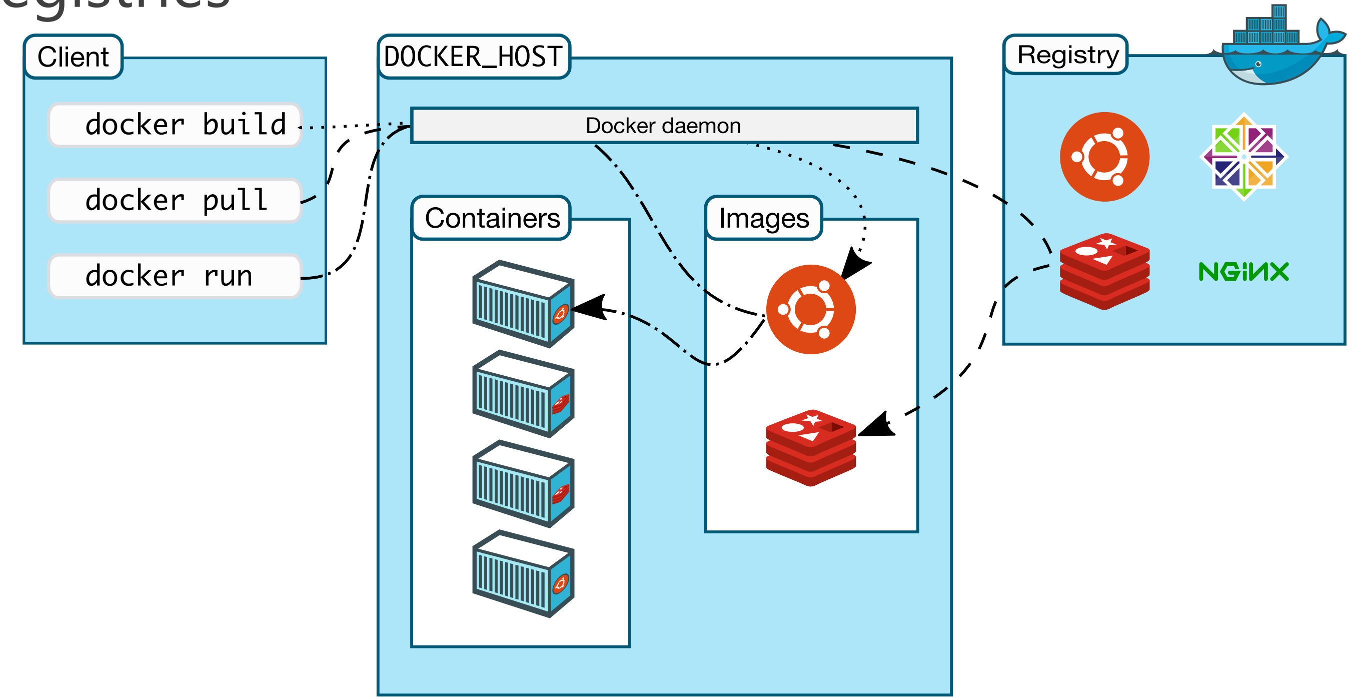
Overview of Docker

- An Open platform for Developing, Shipping and Running Application container.
  - Develop application and its supporting components using containers
  - Container as Unit for distributing and Testing application.
  - Deploy Container into production environment.

SYNERGETICS
GET IT RIGHT

Docker Overview

- Fast, consistent delivery of your applications
- Responsive deployment and scaling
- Higher density than virtual machines
- Image registries

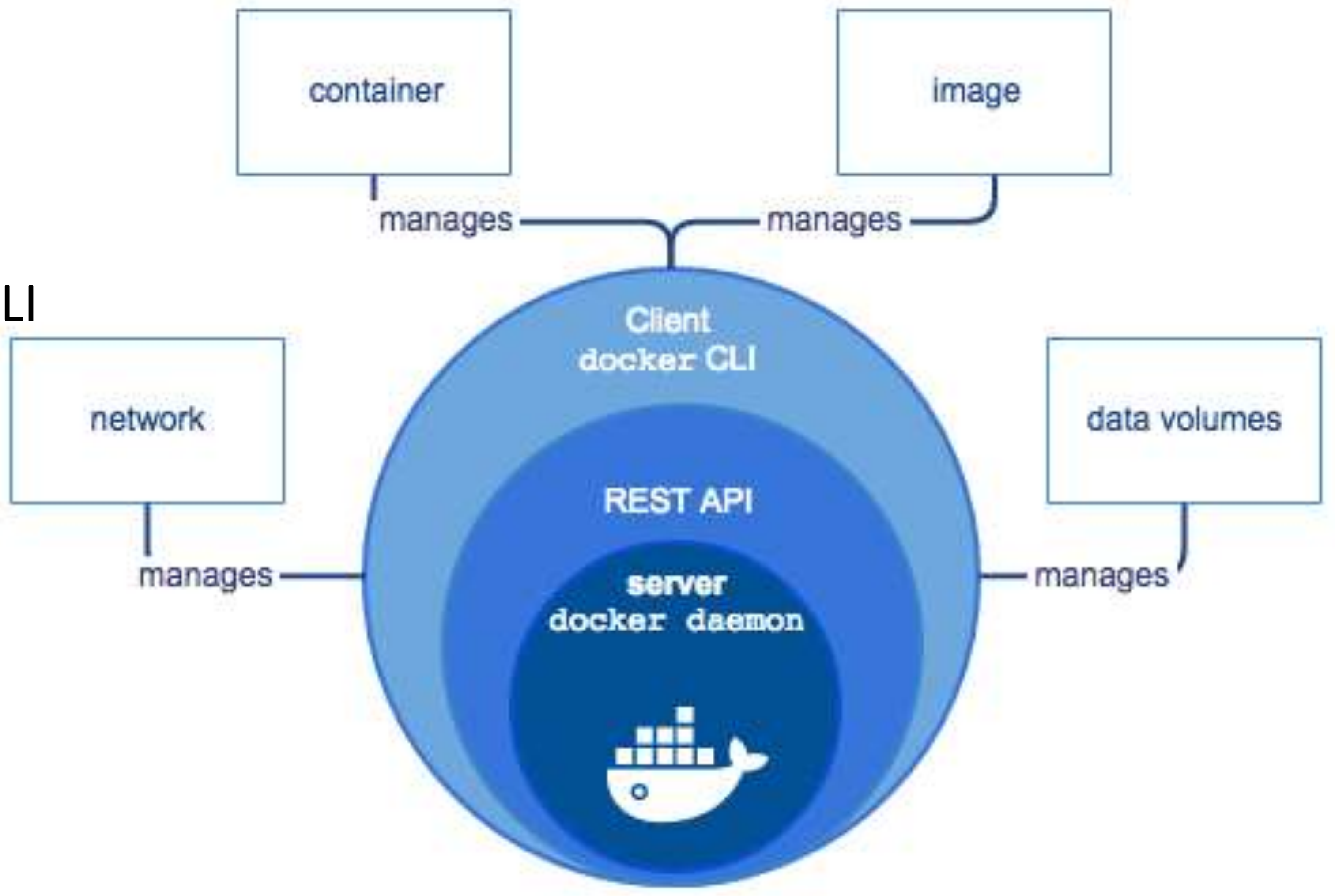


Docker Architecture

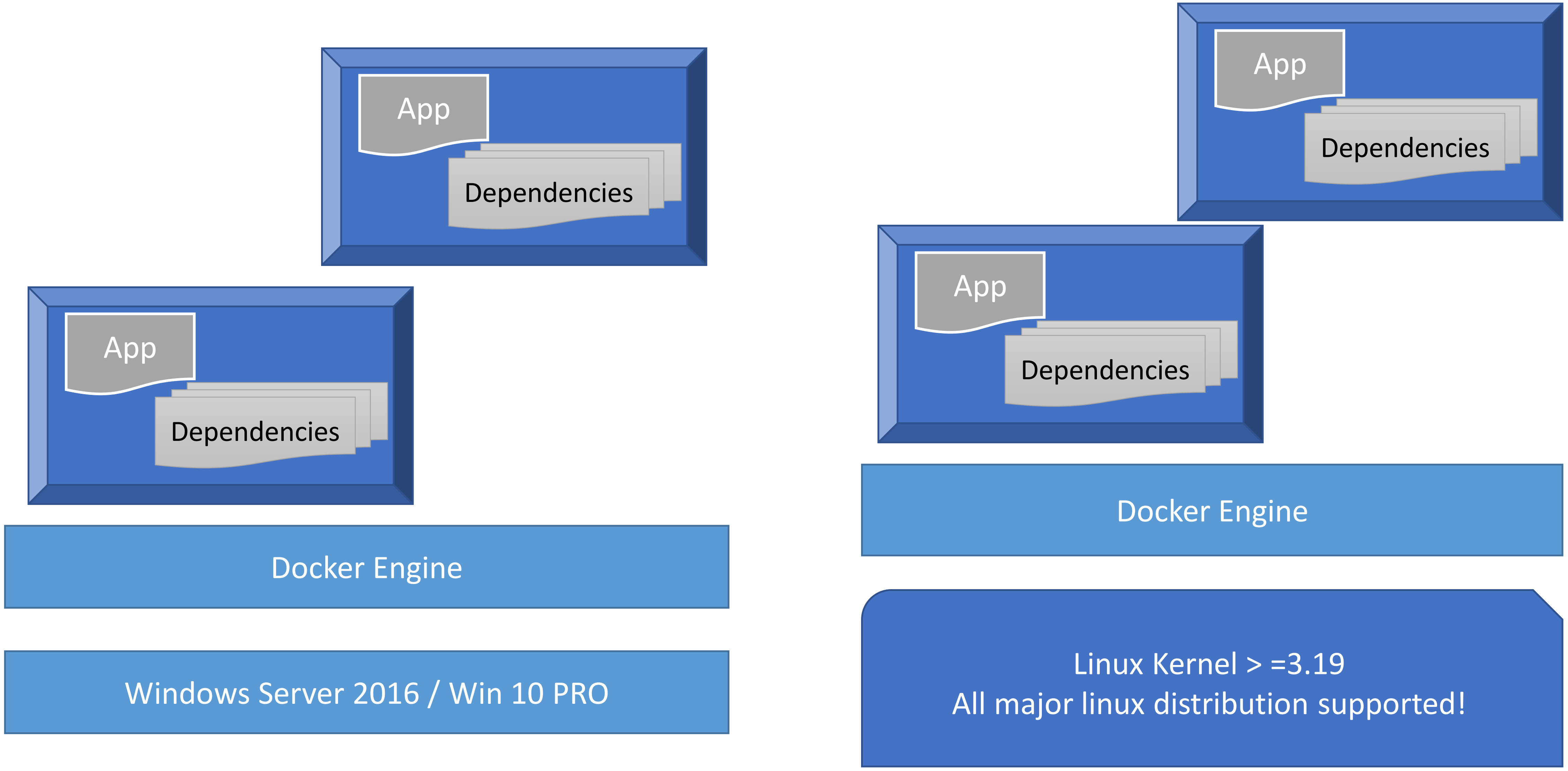
Docker Engine

Is a Client-Server application.

- Docker daemon as Server
- REST API as Interface
  - between daemon and CLI
- CLI client

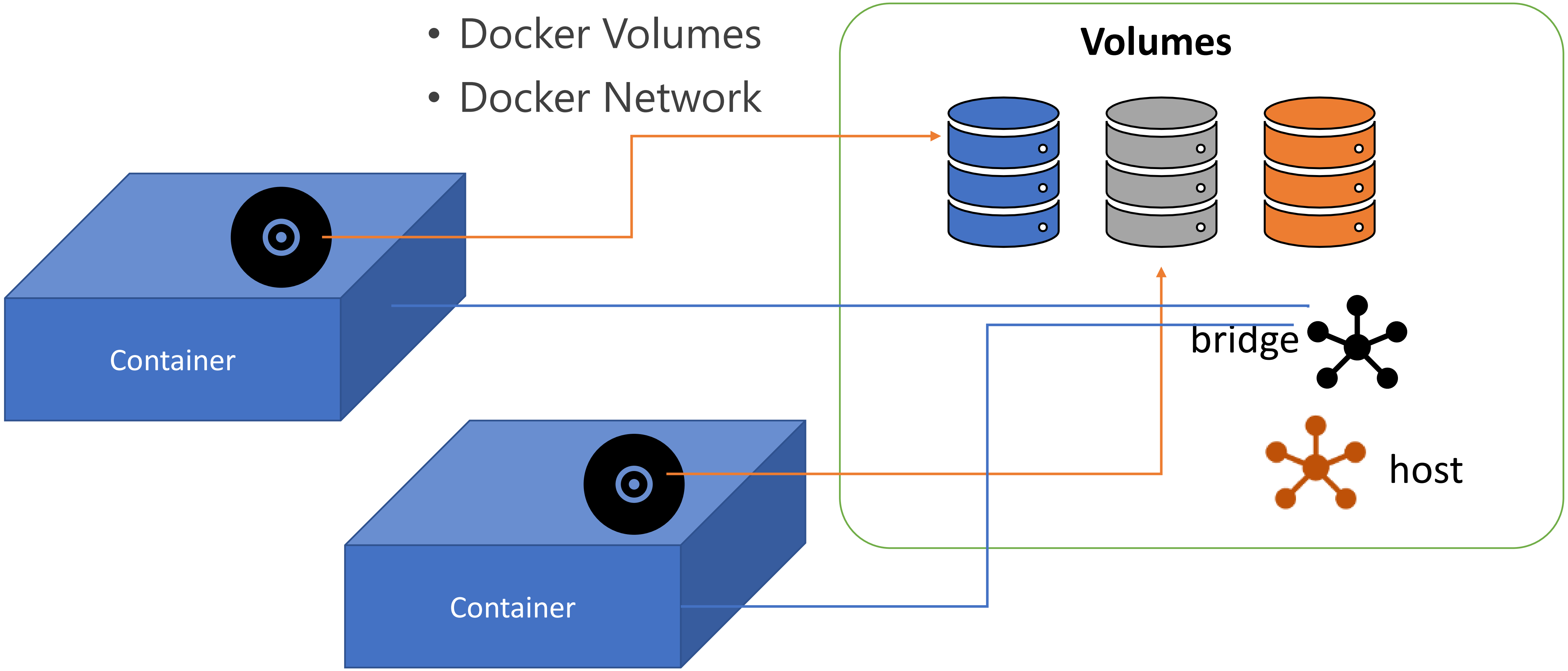


Docker on Windows & Linux



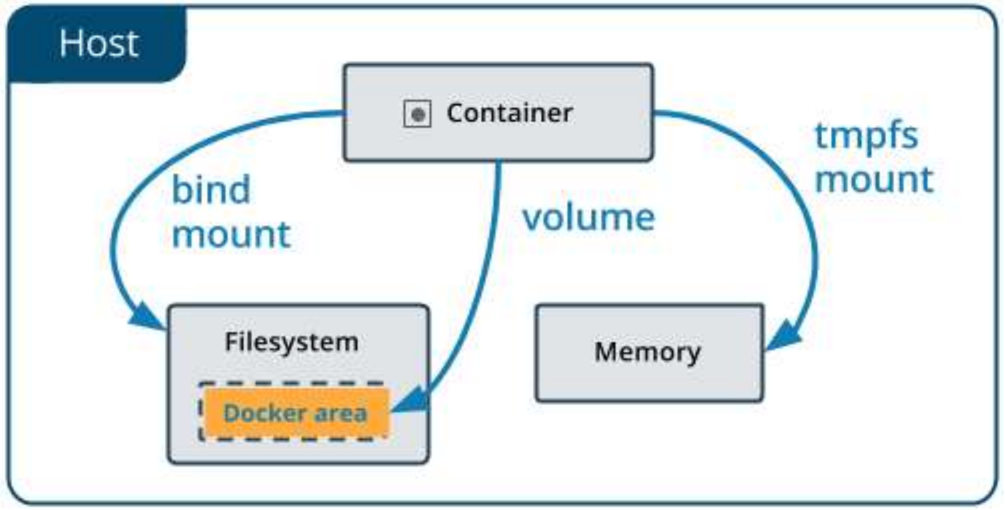
Docker architecture

- Docker Volumes
- Docker Network



Docker Architecture: Volumes

- Volumes
  - Preferred data persisten
  - Managed by docker.
  - Easier to backup or migr
  - Manage using CLI comm
  - Can safely shared amon



Docker Architecture: Networking

- Default Networks

Network Type	Adapter Name	Description
Bridge	Docker0	Default Network, Add containers to Host network.
Host		Add container to Host system only. No Network access.
None		Disable Networking.

Managing Containers with docker

- Docker CLI Commands
  - Images commands
  - Containers commands
  - Other Commands
- Demo: 01 Creating Windows / Linux Container with Web Server
- NOTE: This demo doesn’t include any sample page. You should get your Web Server’s default welcome page.

Automating Container build

- Dockerfile and it’s syntax
- Building a new container and image using Dockerfile

- NOTE: This demo uses simple HTML page. No Server side programming needed.

Docker Repositories

- The Registry is a stateless, highly scalable server side application that stores and lets you distribute Docker images.
- Allows sharing of images.
- Docker can pull and push images from repository.
- Repository type:
  - Local repository
    - A Special Container from Image “registry”
    - Not secure, need TLS for security
  - Dockerhub repository
    - A cloud based registry available on subscription basis.
    - Integration with docker cli.

Dockerhub demo

- Demo 03: Signup for dockerhub.
- Demo 04: Push your local images to dockerhub.

SYNERGETICS GET IT RIGHT | Microsoft Partner Gold Cloud Platform Silver Learning

Q & A