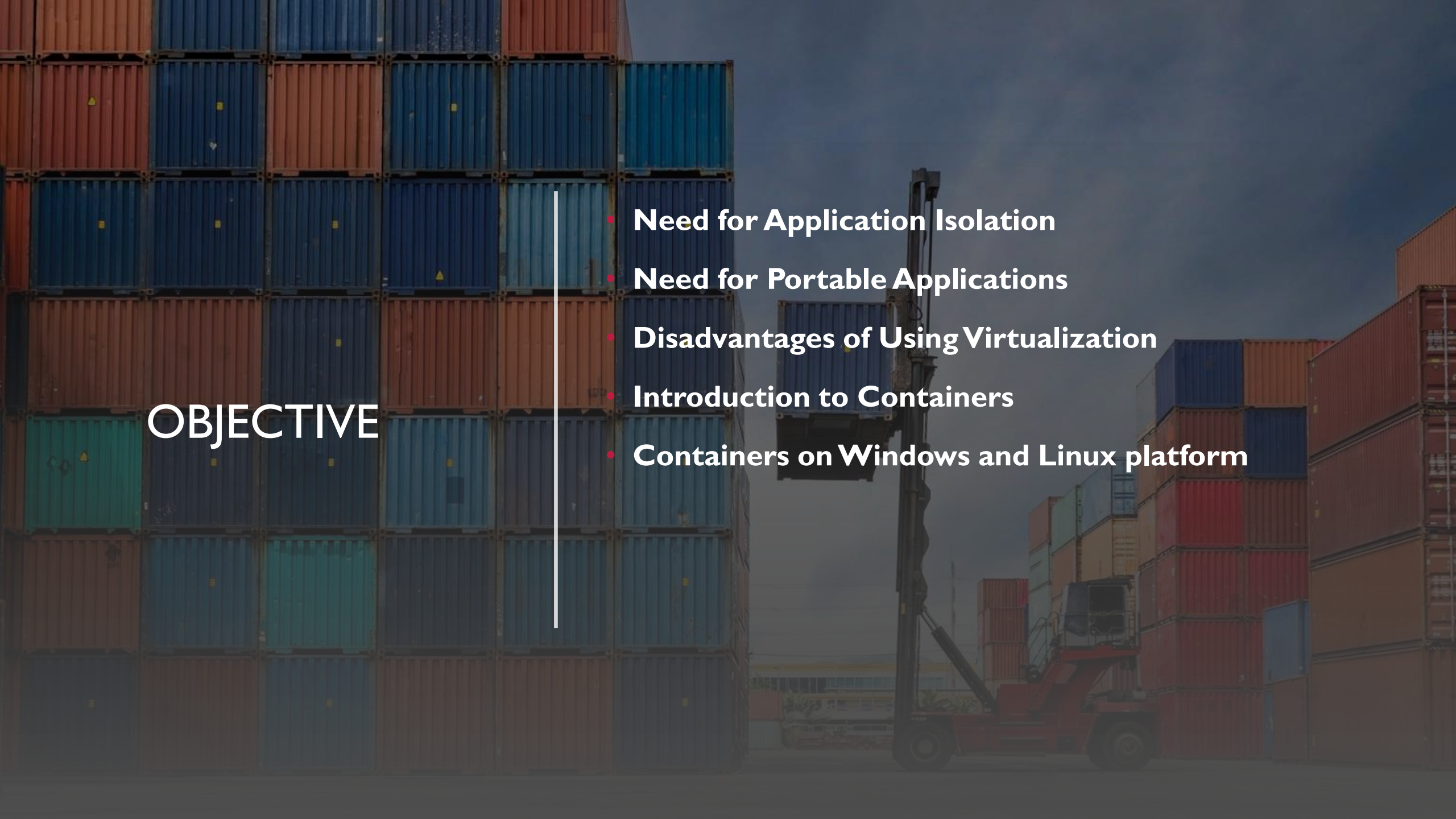
CONTAINERIZATION WITH DOCKER

INTRODUCTION TO CONTAINERS

MODULE 1



OBJECTIVE

- Need for Application Isolation
- Need for Portable Applications
- Disadvantages of Using Virtualization
- Introduction to 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.

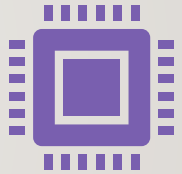
Application should have its own sandbox.

NEED FOR PORTABLE APPLICATIONS

Application goes through following environments:

Development
Testing
Staging
Production

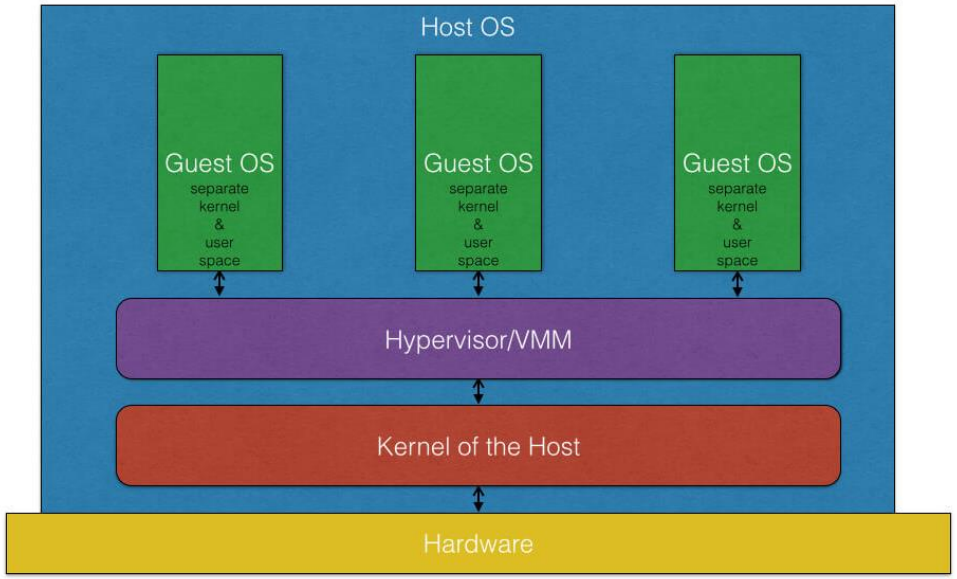
Managing Dependencies across all environments could be difficult.



Creating a compatible dev-test environment may take considerable time.

VIRTUALIZATION

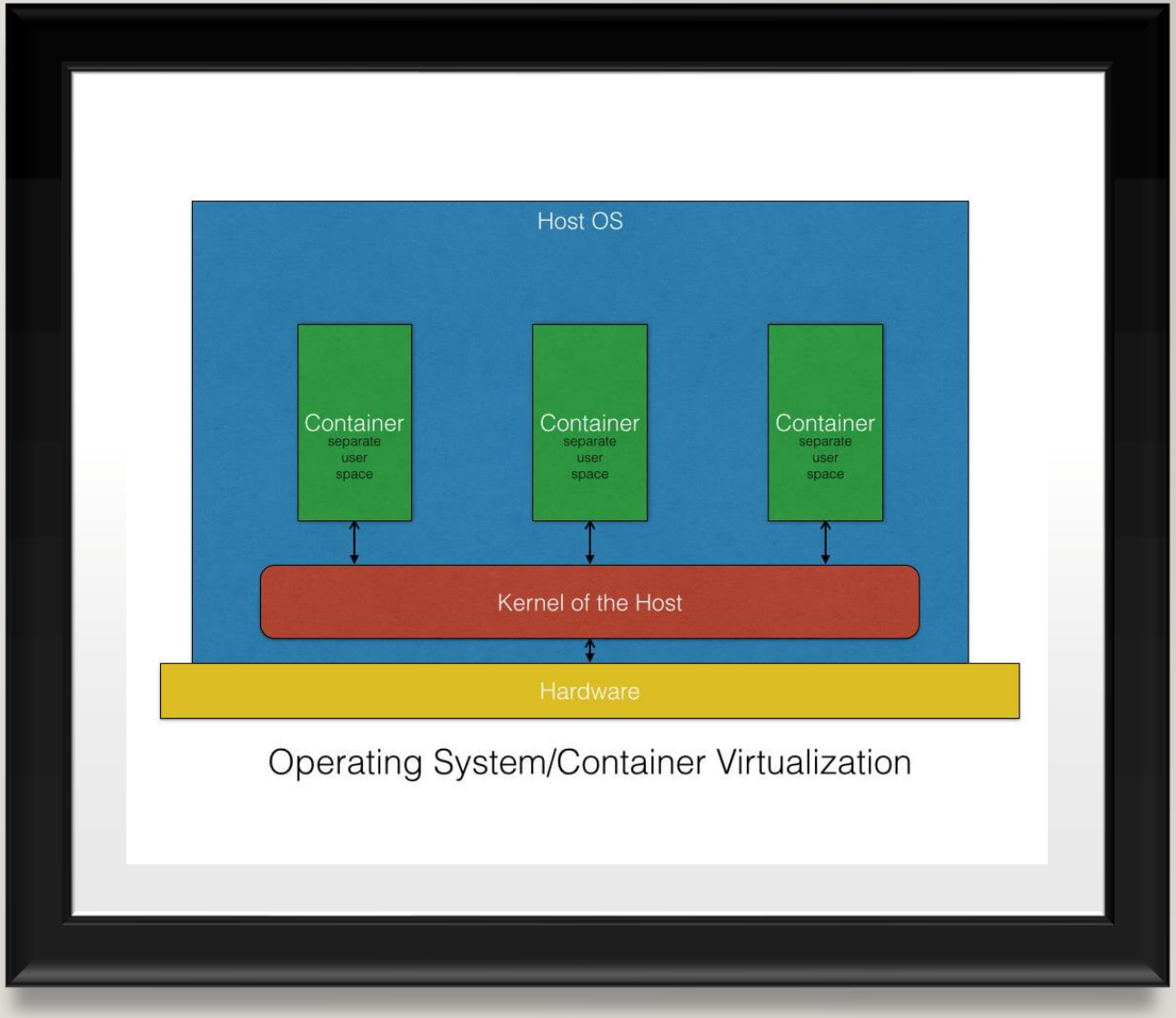
- Complete Isolation [Virtualize Hardware and Operating System]
- Time Consuming
- Not ideal for isolating Individual application.



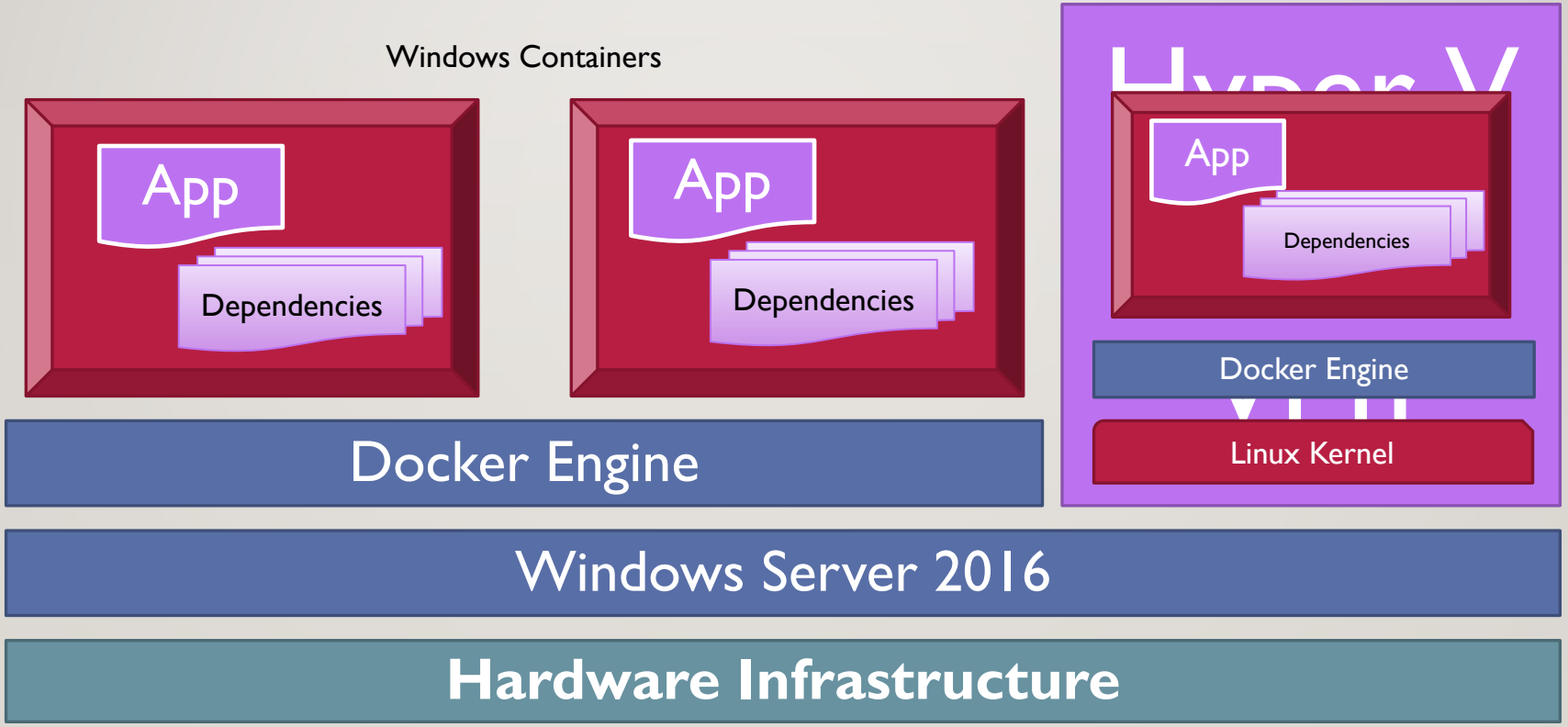
Hypervisor based Virtualization

INTRODUCTION TO CONTAINERS

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



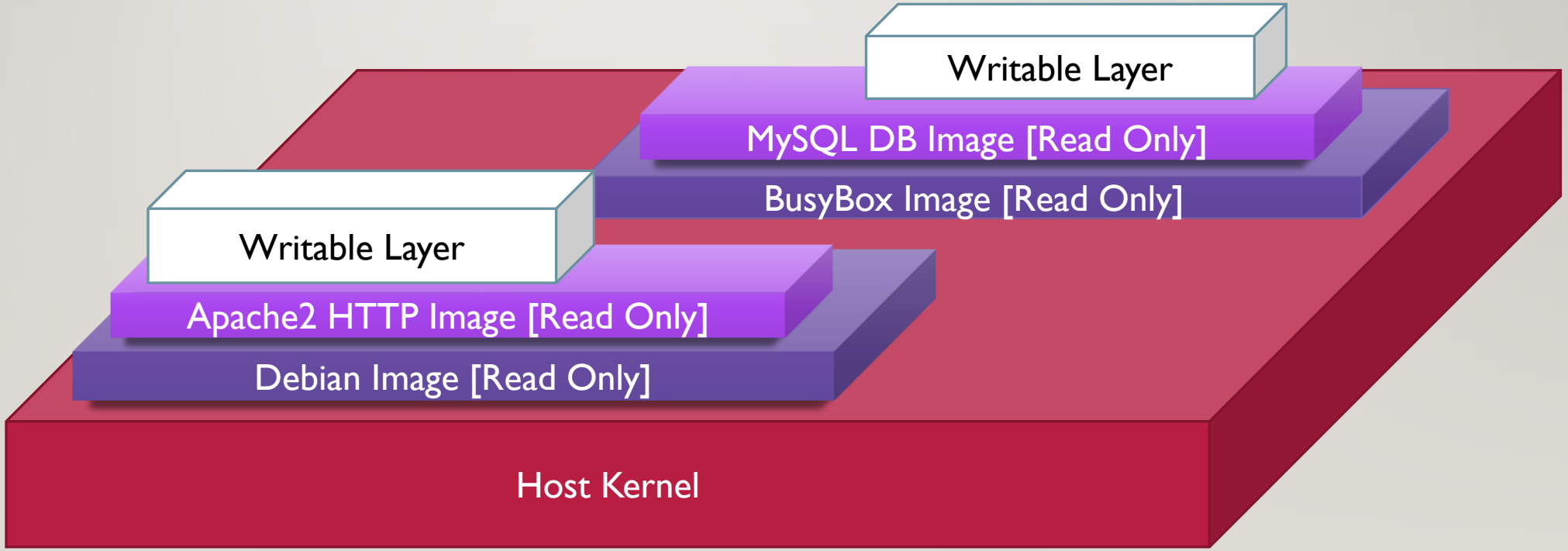
CONTAINERS FOR WINDOWS AND LINUX



CONTAINERS IN DETAILS

Module 2

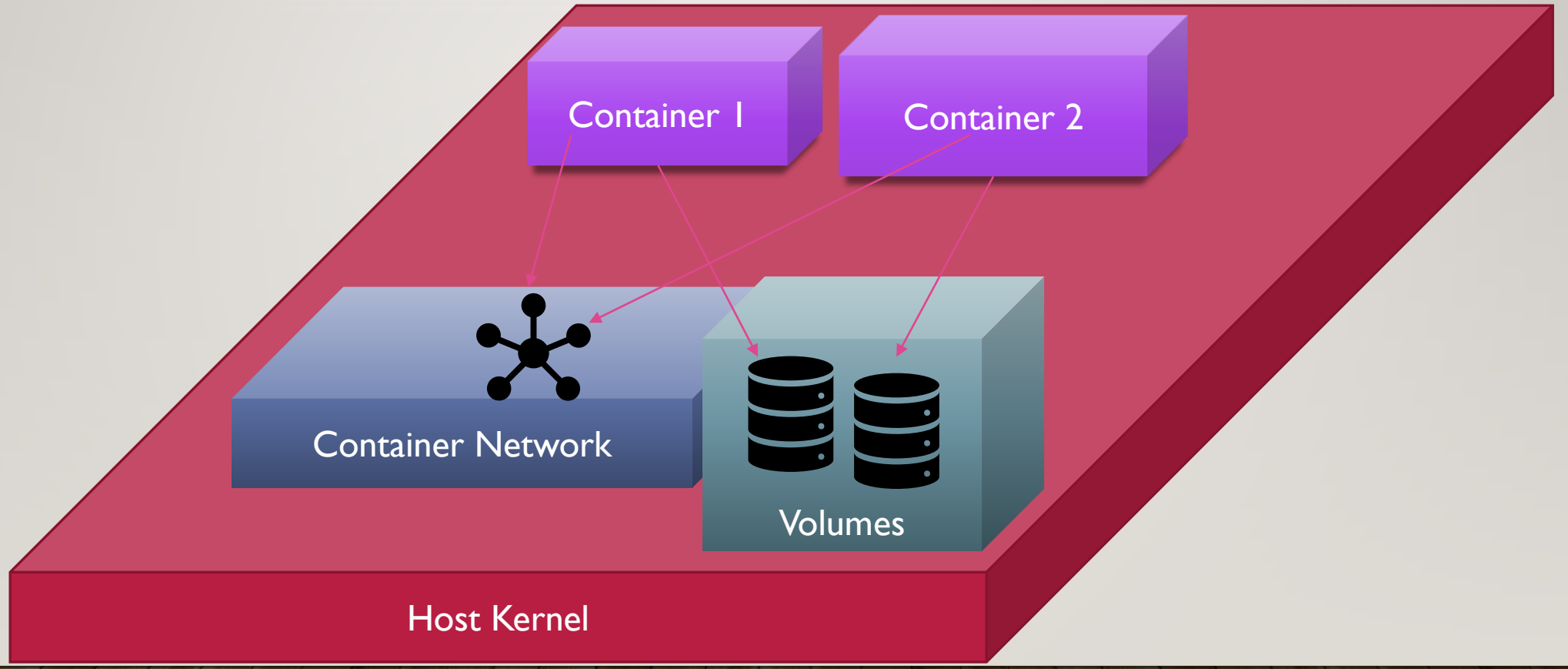
CONTAINER ARCHITECTURE



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.

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.

INTRODUCTION TO DOCKER

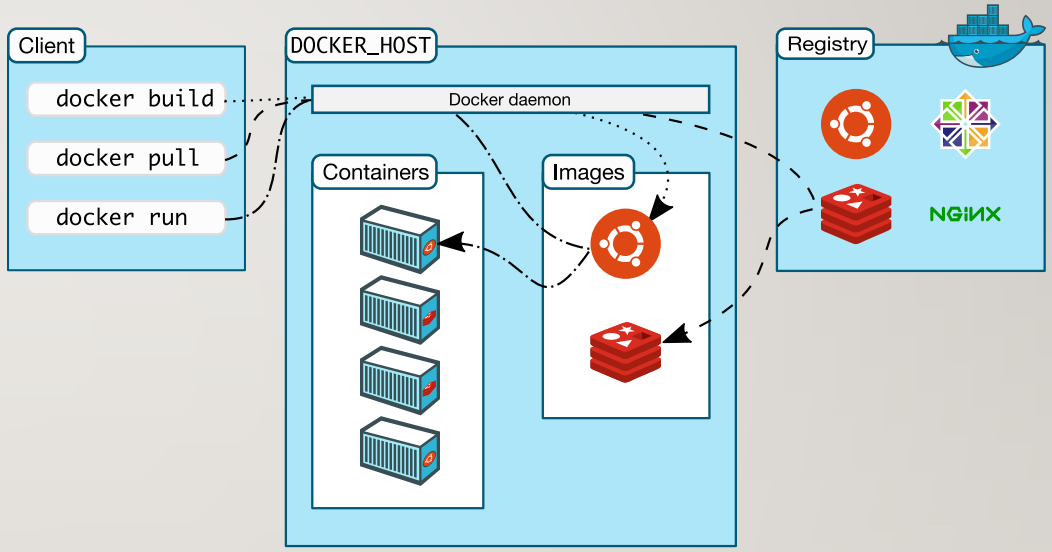
Module 3

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.

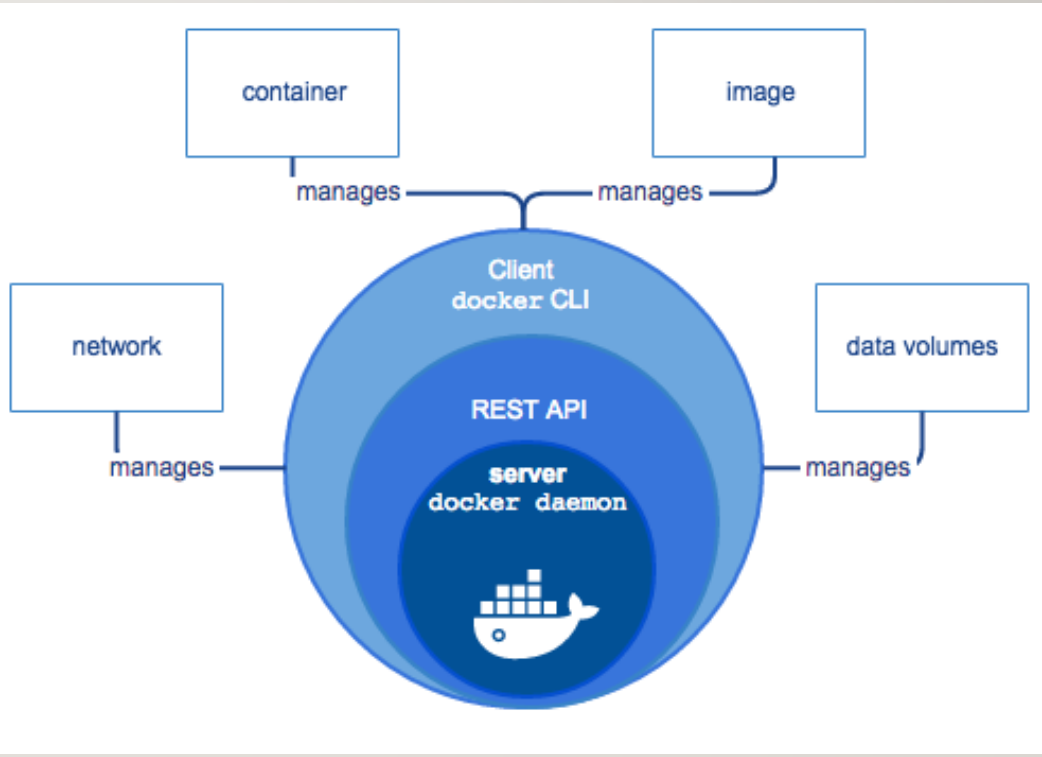
DOCKER OVERVIEW

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

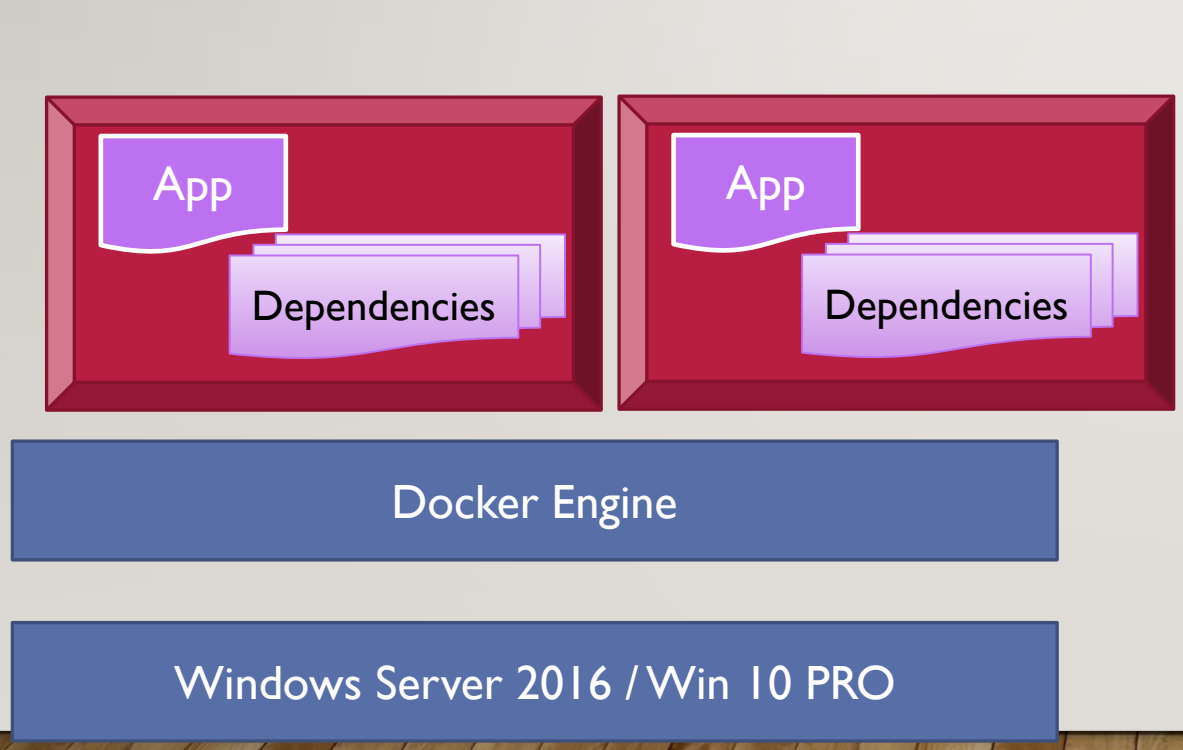


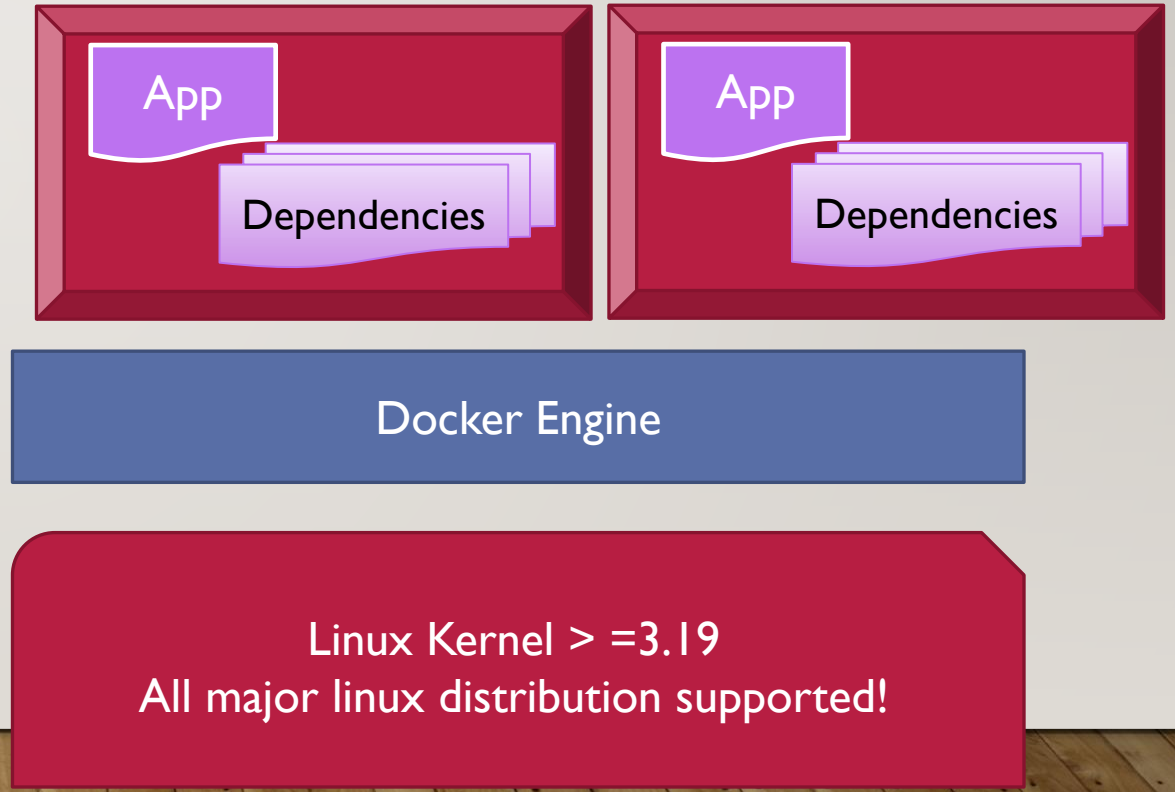
DOCKER ARCHITECTURE

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



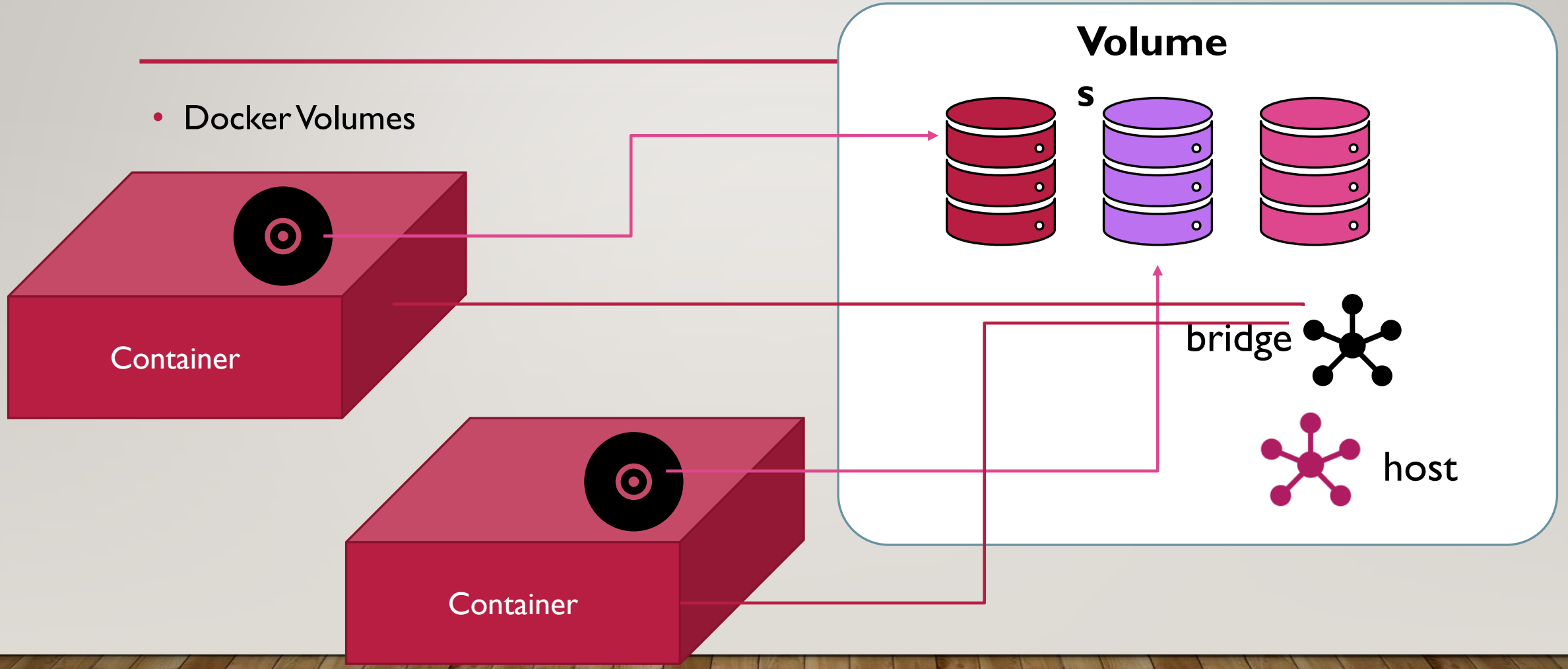
DOCKER ON WINDOWS & LINUX





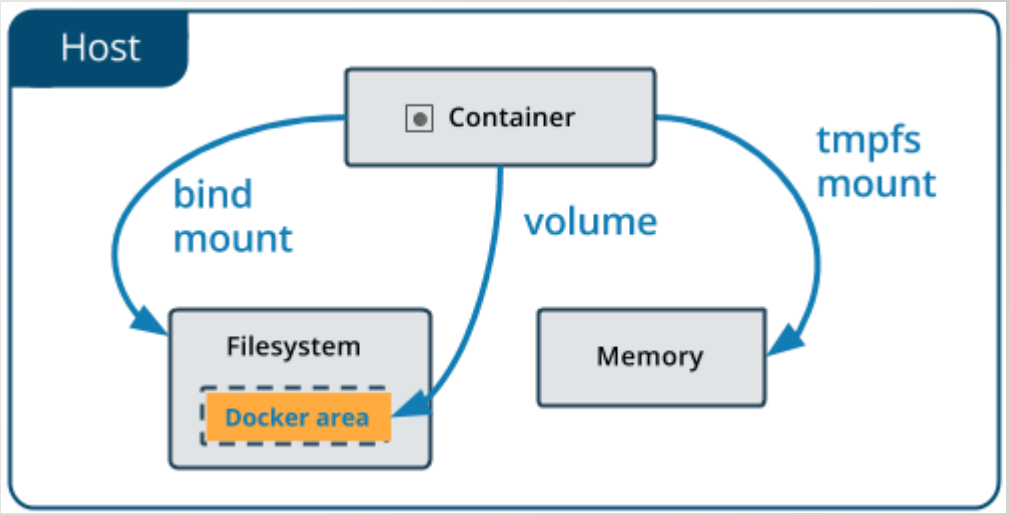
DOCKER ARCHITECTURE

- Docker Volumes



DOCKER ARCHITECTURE: VOLUMES

- Volumes
  - Preferred data persistence mechanism.
  - Managed by docker.
  - Easier to backup or migrate.
  - Manage using CLI commands.
  - Can safely shared among containers.



DOCKER ARCHITECTURE: NETWORKING

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 Container which prints "Hello World!"

AUTOMATING CONTAINER BUILD

- Dockerfile and it's syntax
- Building a new container and image using Dockerfile
- Demo 02: Creating a new container with Java Console Based application.

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.

Q & A