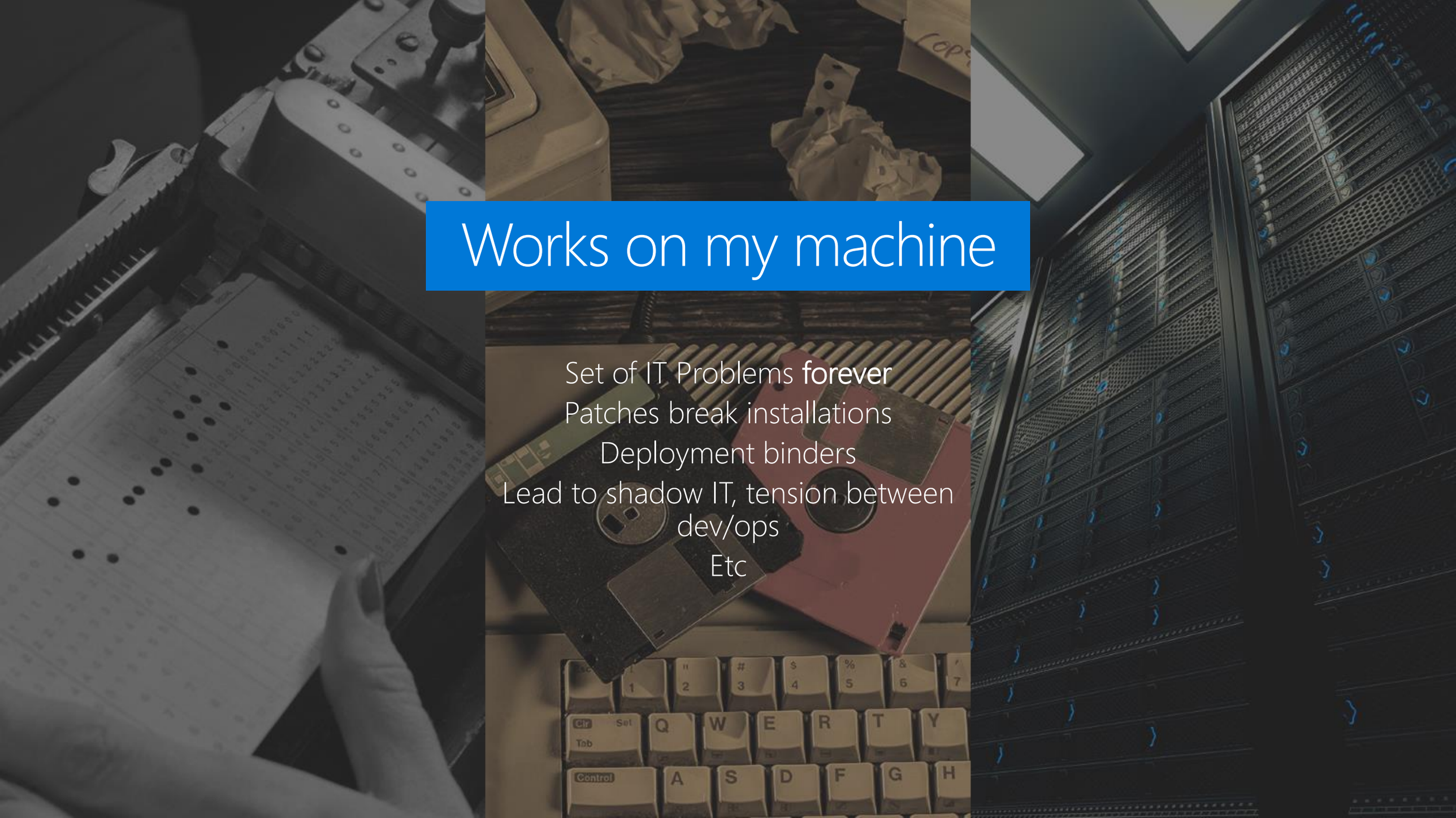
Microsoft

Containers

Miguel Caldas
Principal Technical Evangelist



Works on my machine

Set of IT Problems **forever**

Patches break installations

Deployment binders

Lead to shadow IT, tension between dev/ops

Etc

The cloud has changed expectations

Availability

100% Uptime

Hyper-scale

From startup to enterprise

Agility

Deliver just in time speed

What problems do containers solve?

Containers deliver speed, flexibility, and savings

Availability

62%

Report reduction in MTTR

10X

Cost reduction in maintaining existing applications

Hyper-scale

41%

Move workloads across private/public clouds

Eliminate

"works on my machine" issues

Agility

13X

More software releases

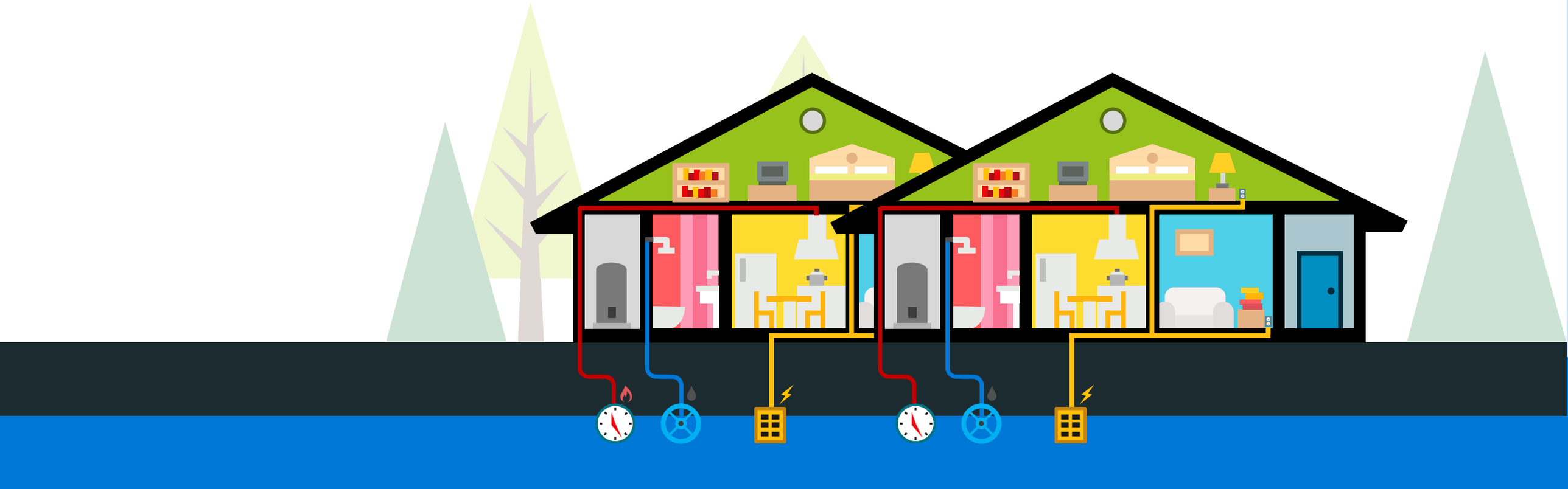
65%

Reduction in developer onboarding time

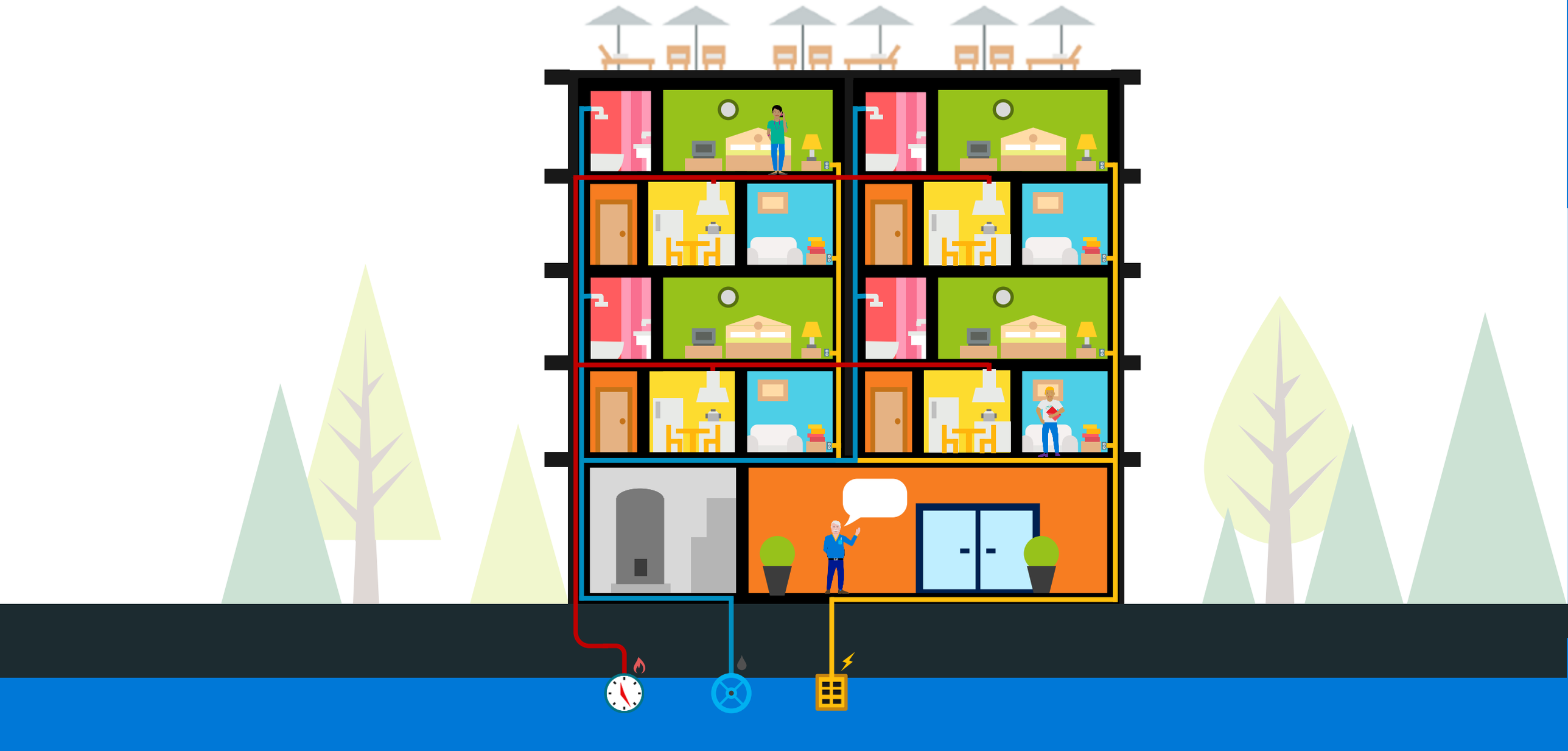
State of App development Survey: Q1 2016, Cornell University case study

What is a container?

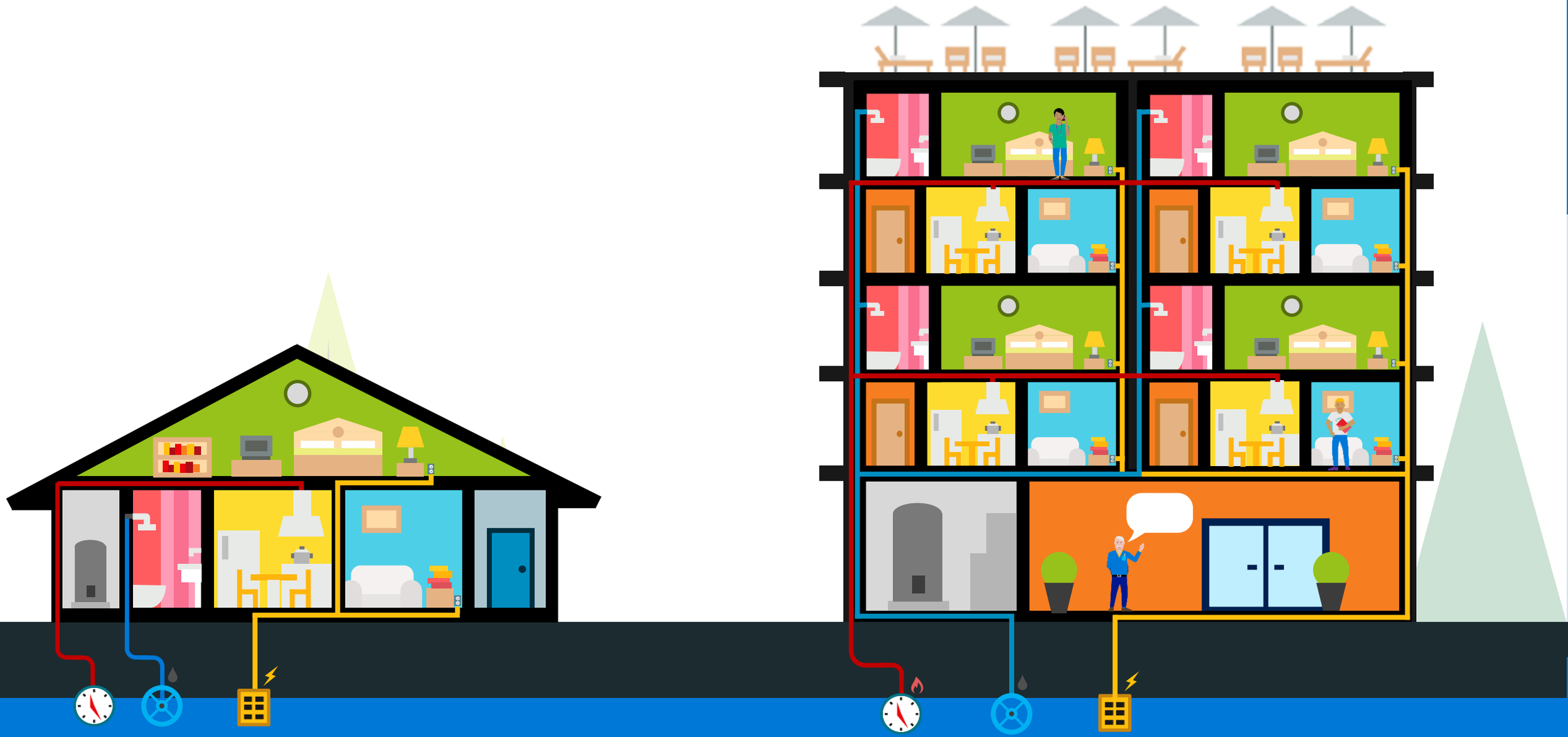
Virtual Machine (VM) vs. Container



Virtual Machine (VM) vs. Container

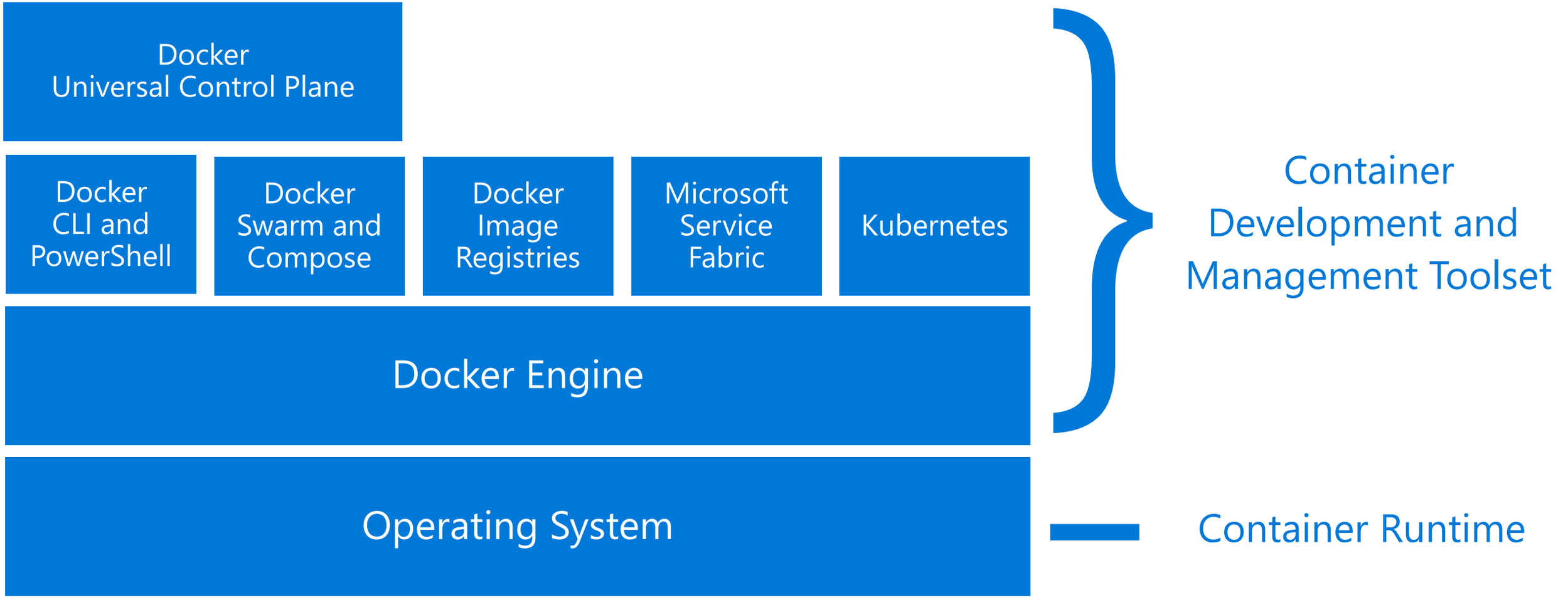


Virtual Machine (VM) vs. Container

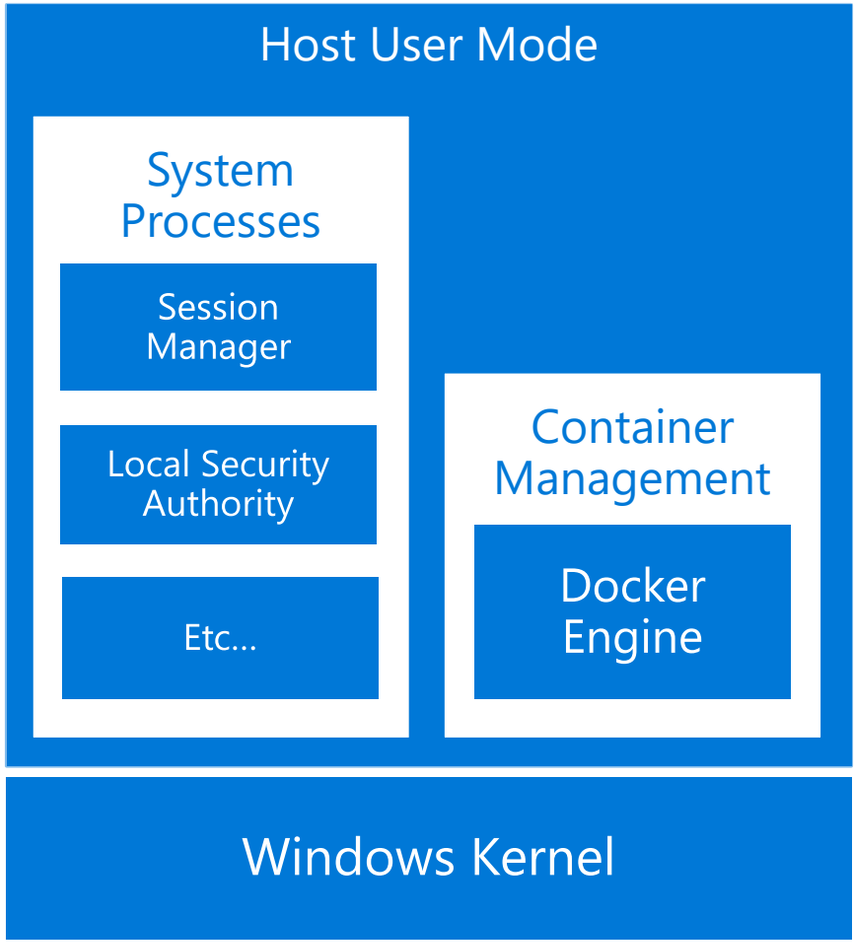


How do containers work?

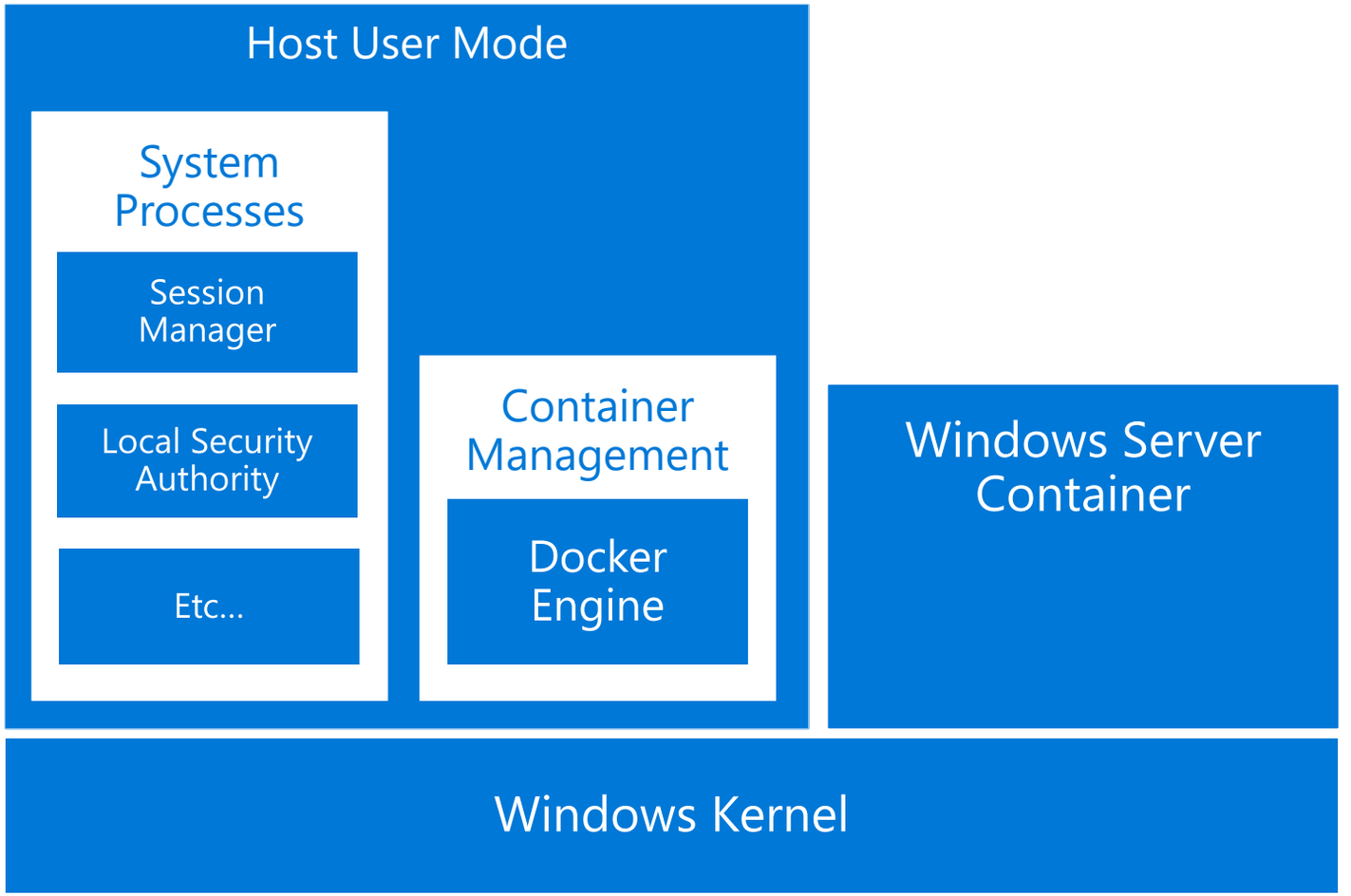
High Level Architecture



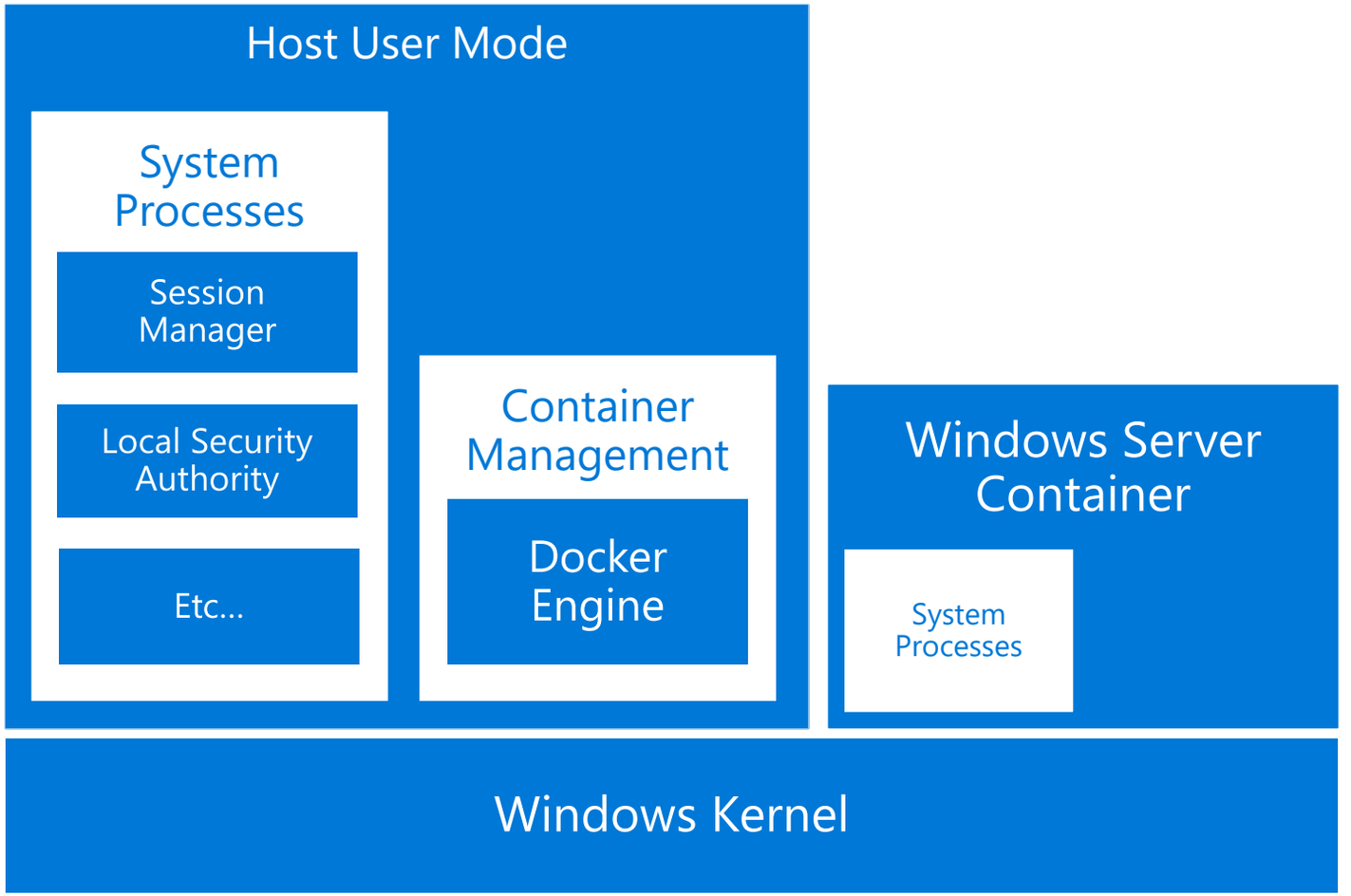
Windows Containers



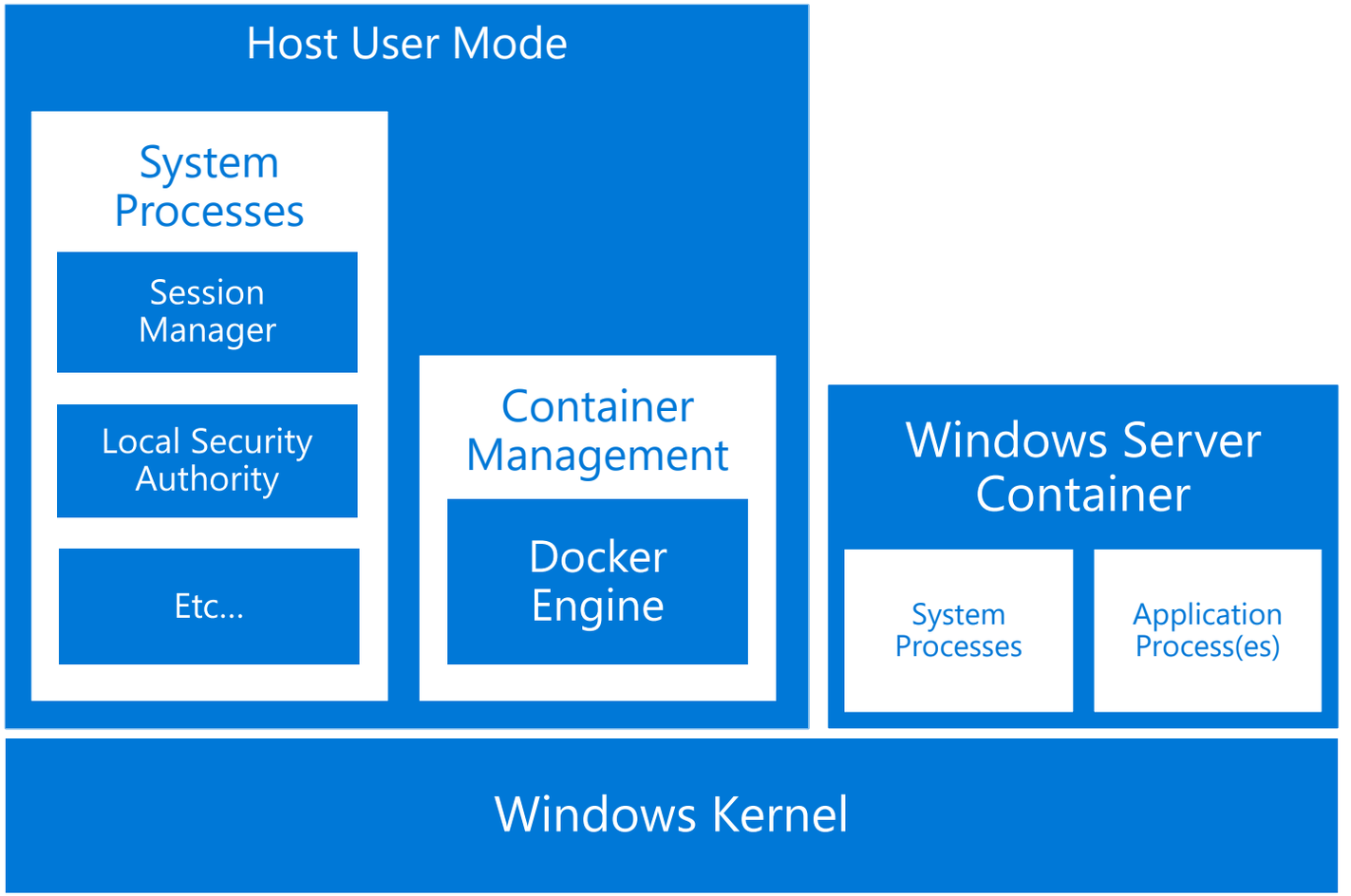
Windows Containers



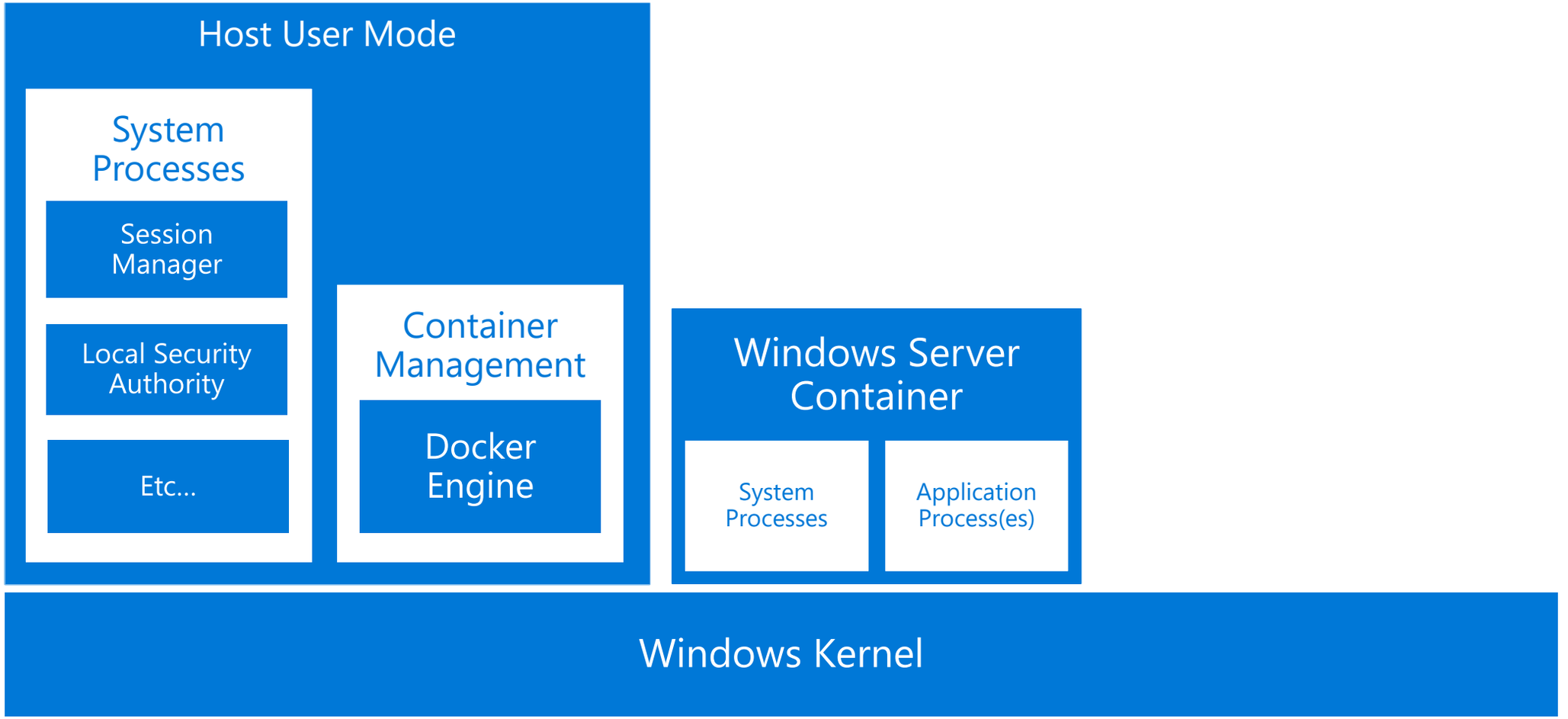
Windows Containers



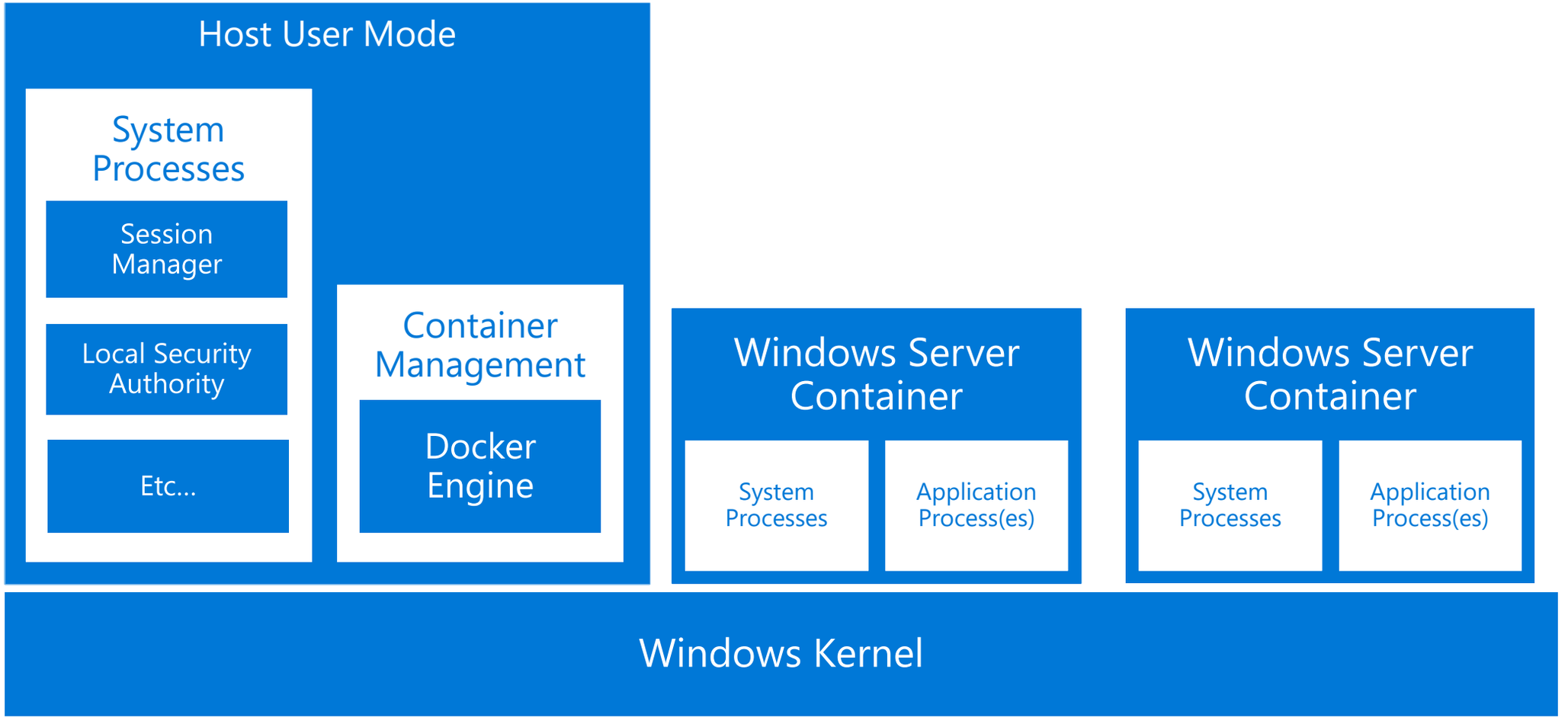
Windows Containers



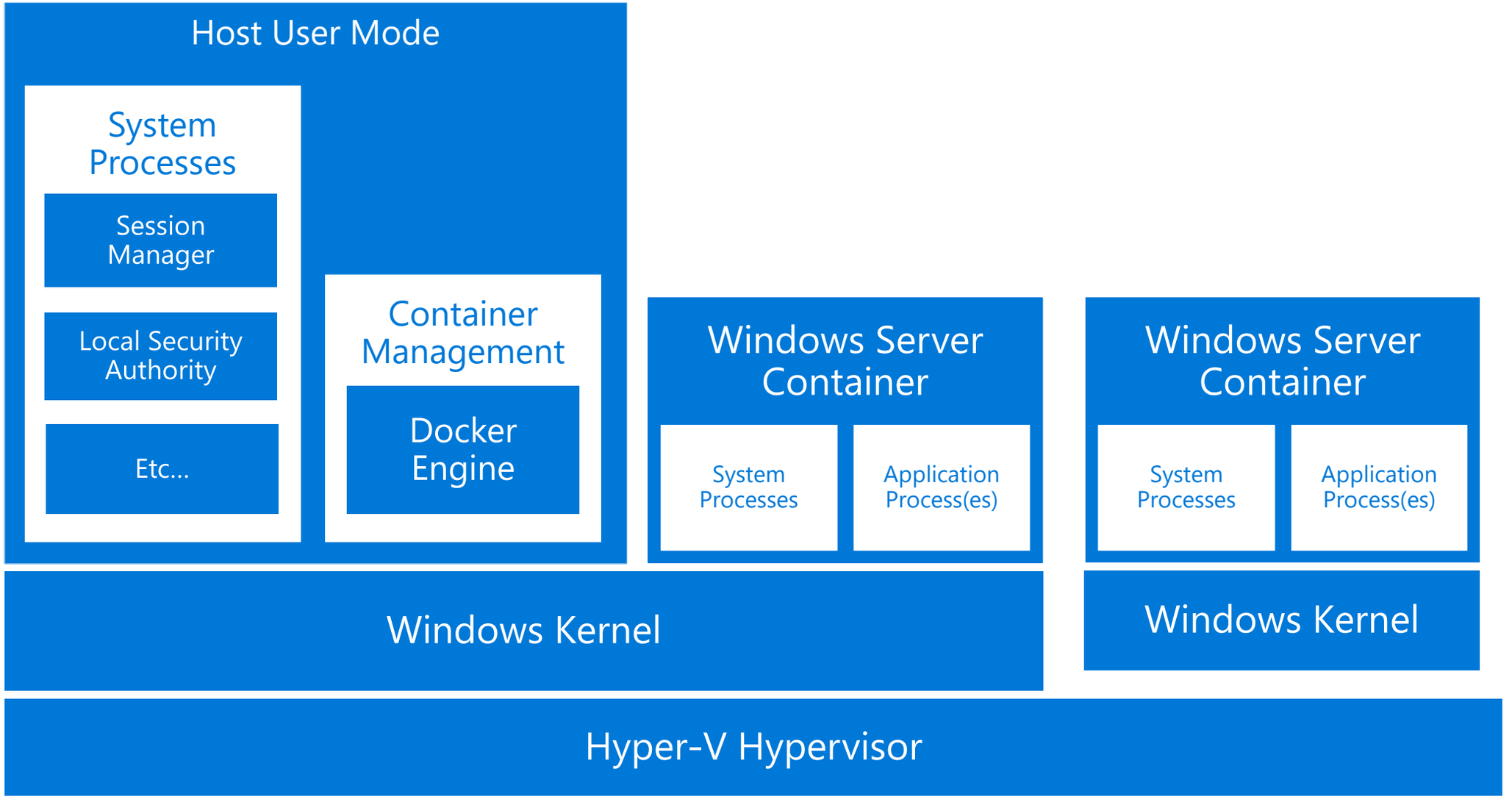
Windows Containers



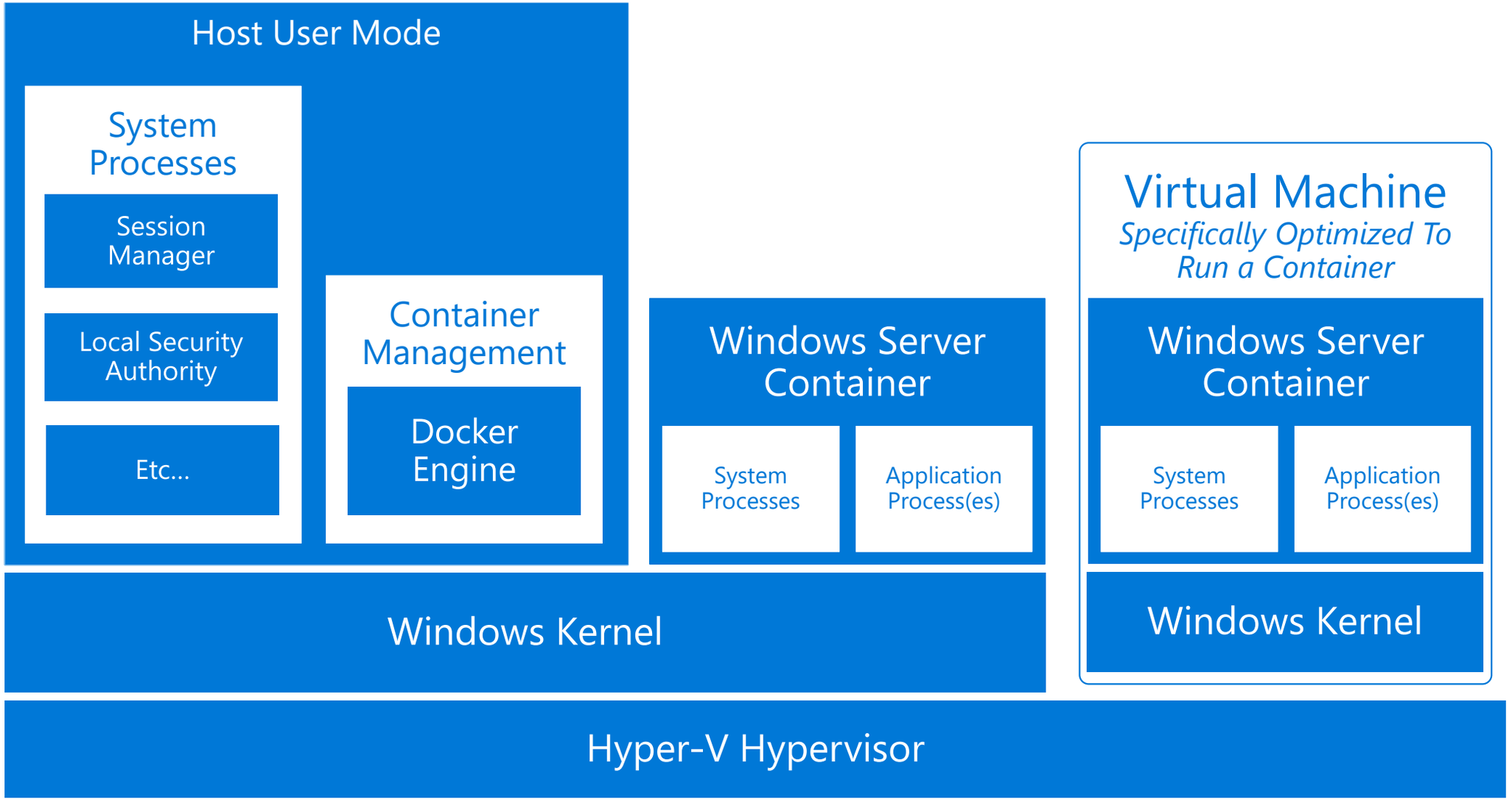
Windows Containers



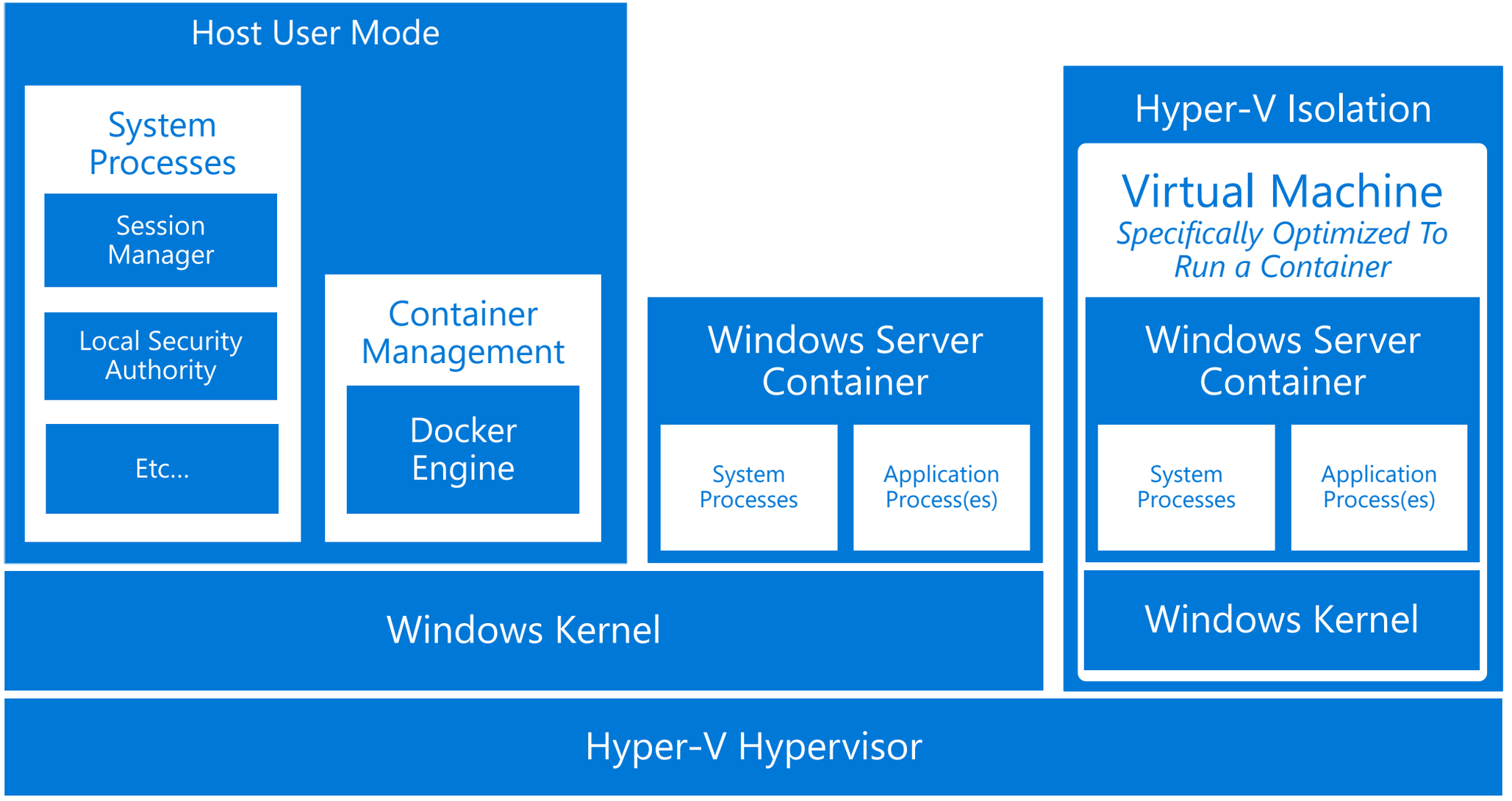
Windows Containers



Windows Containers



Windows Containers



Startup Performance

NodeJS with Windows Server Core

Windows Server Container
~1 second

With Hyper-V Isolation
~3.3 seconds

A virtual machine takes ~**5 seconds to over a min**

*Includes initial boot provisioning phase (out-of-box setup etc…)

**Startup time after initial container start.

Testing performed on HP ProLiant SL250s Gen8, E5-2600, 2 Socket, 8 Core, 128GB RAM, HP SATA SSD - results may vary based on hardware and software configurations.

Startup Performance

NodeJS with Nano Server

Windows Server Container
Under 600 Milliseconds!

With Hyper-V Isolation
~1.75 seconds

A virtual machine takes ~**3 seconds**

NodeJS with Windows Server Core

Windows Server Container
~1 second

With Hyper-V Isolation
~3.3 seconds

A virtual machine takes ~**5 seconds to over a min**

Density

NodeJS with Windows Server Core

Windows Server Container

First Container **~150MB**
Additional Containers **~75MB**

With Hyper-V Isolation

First Container **~555MB**
Additional Containers **~280MB**

Testing performed on HP ProLiant SL250s Gen8, E5-2600, 2 Socket, 8 Core, 128GB RAM, HP SATA SSD - results may vary based on hardware and software configurations.

Density

NodeJS with Nano Server

Windows Server Container

First Container **~120MB**
Additional Containers **~75MB**

With Hyper-V Isolation

First Container **~340MB**
Additional Containers **~150MB**

NodeJS with Windows Server Core

Windows Server Container

First Container **~150MB**
Additional Containers **~75MB**

With Hyper-V Isolation

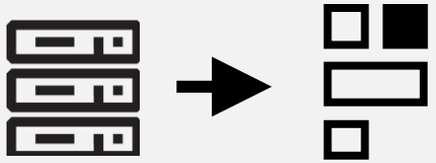
First Container **~555MB**
Additional Containers **~280MB**

Where should I start?

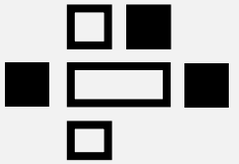
One platform, one journey

CONTAINERIZE TRADITIONAL APPLICATIONS
Containerize for security, portability and efficiency

TRANSFORM MONOLITHIC TO MICROSERVICES
Look for shared services to transform



ACCELERATE NEW APPLICATIONS
Agile cloud native app development

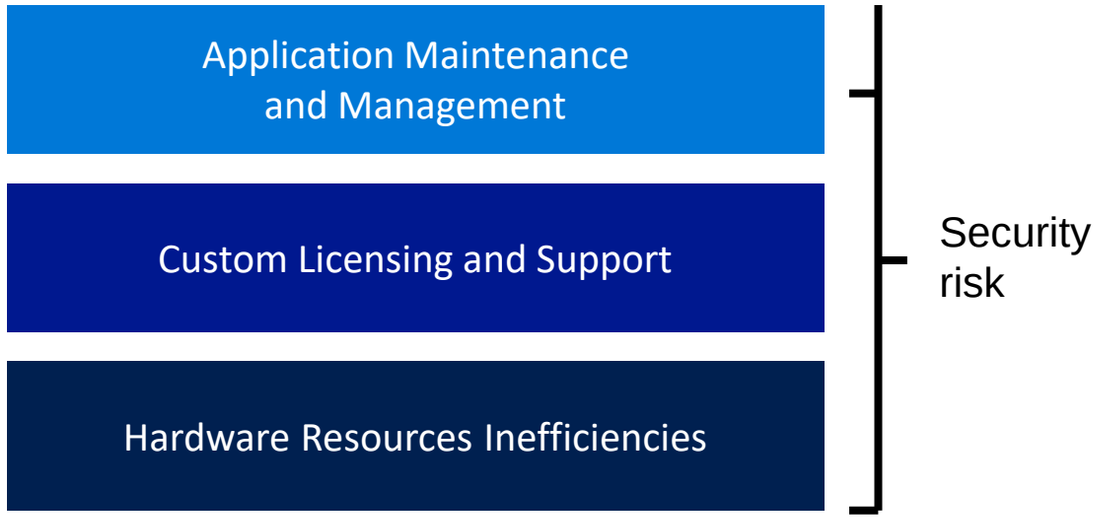


Why containerize traditional Windows apps?

PORTABILITY, SECURITY AND COST REDUCTION

- Accelerate your public and hybrid cloud strategy
- Gain applications portability, agility, and control
- Reduce infrastructure and management costs for Windows 2003, 2008, and 2012 Windows applications

COSTS FOR MANAGING TRADITIONAL APPLICATIONS



Top Development Challenges for Development Teams

65%

Traditional Application Maintenance

59%

Inertia of Traditional App Infrastructure

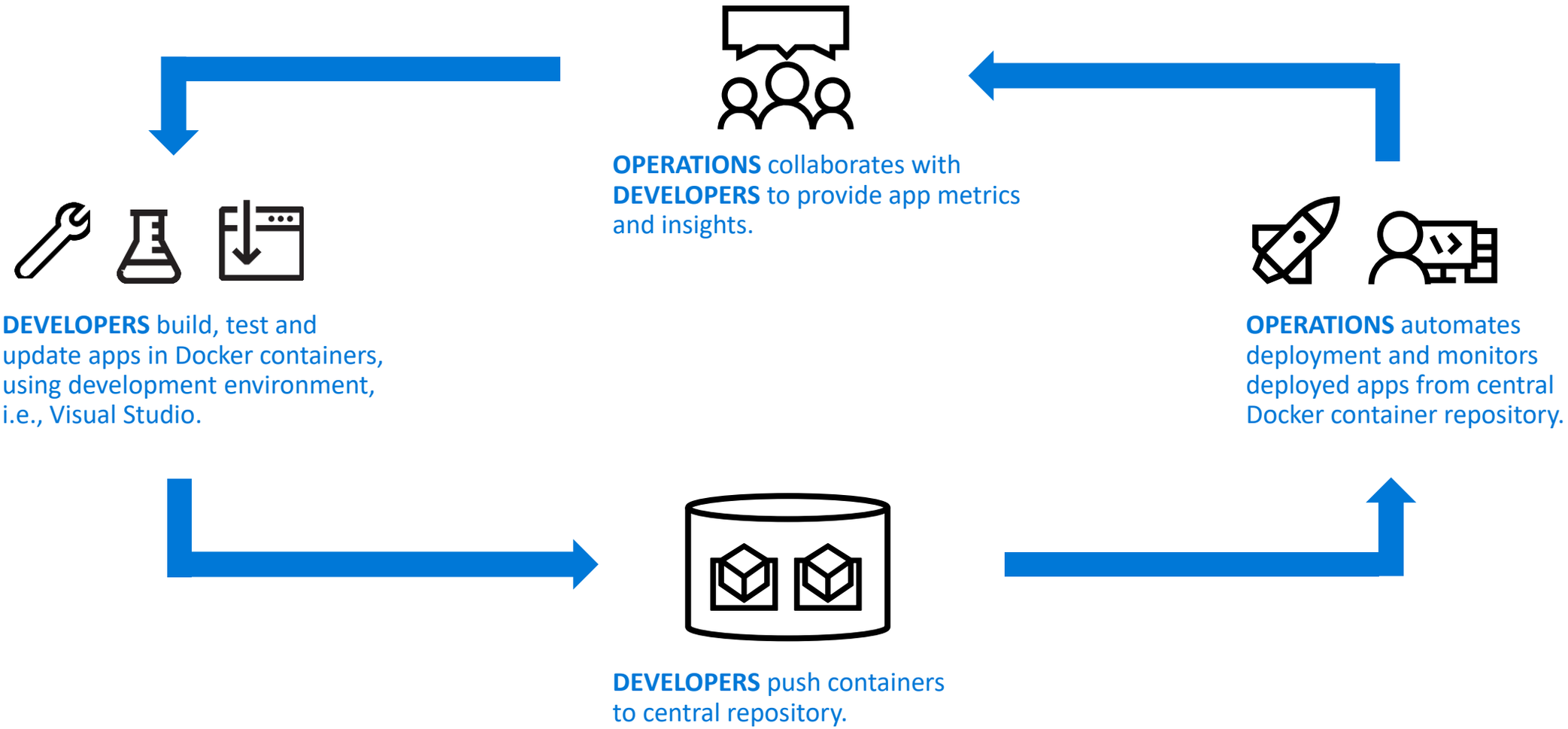
State of App development Survey: Q1 2016, Cornell University case study.

> Most of our application portfolio consists of older legacy applications that are cumbersome to update. By moving these applications into Windows Server Containers with Docker and embracing a microservices architecture, we can break these big applications apart and update the pieces independently. This will reduce customer downtime and increase business agility.

STEPHEN TARMEY
Chief Architect
Tyco International

Collaborative DevOps Approach to apps

Docker containers are central to process



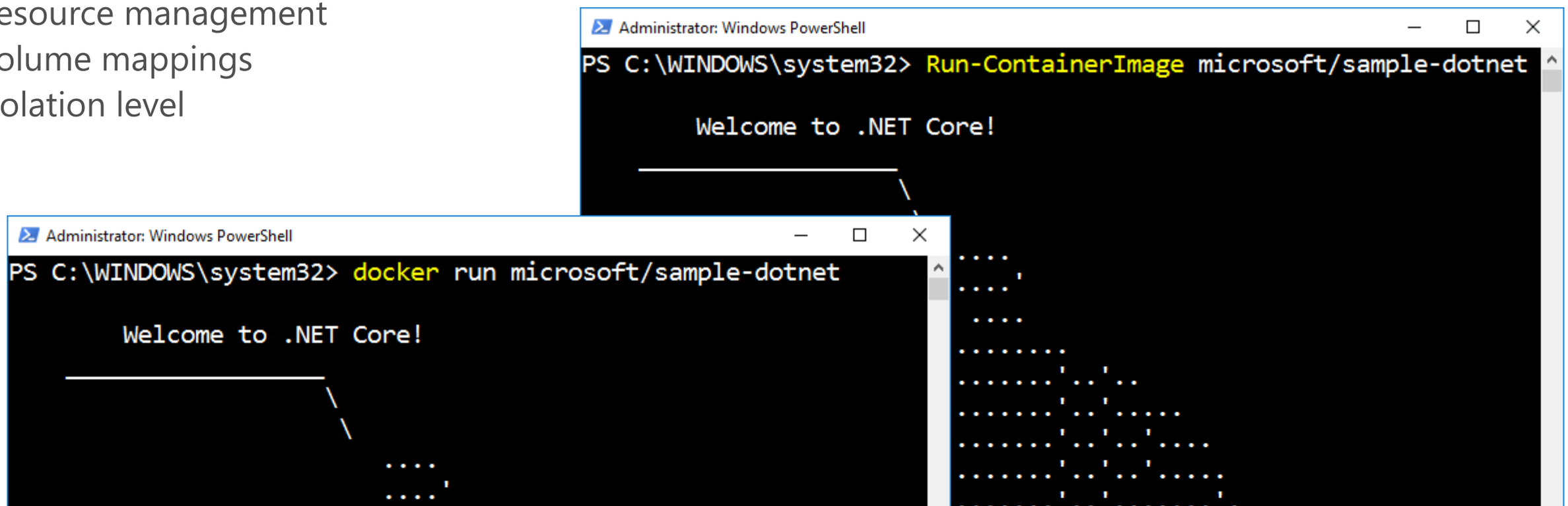
How do I actually create a container?

Docker Run

Creates and Starts a New Container

Runtime options

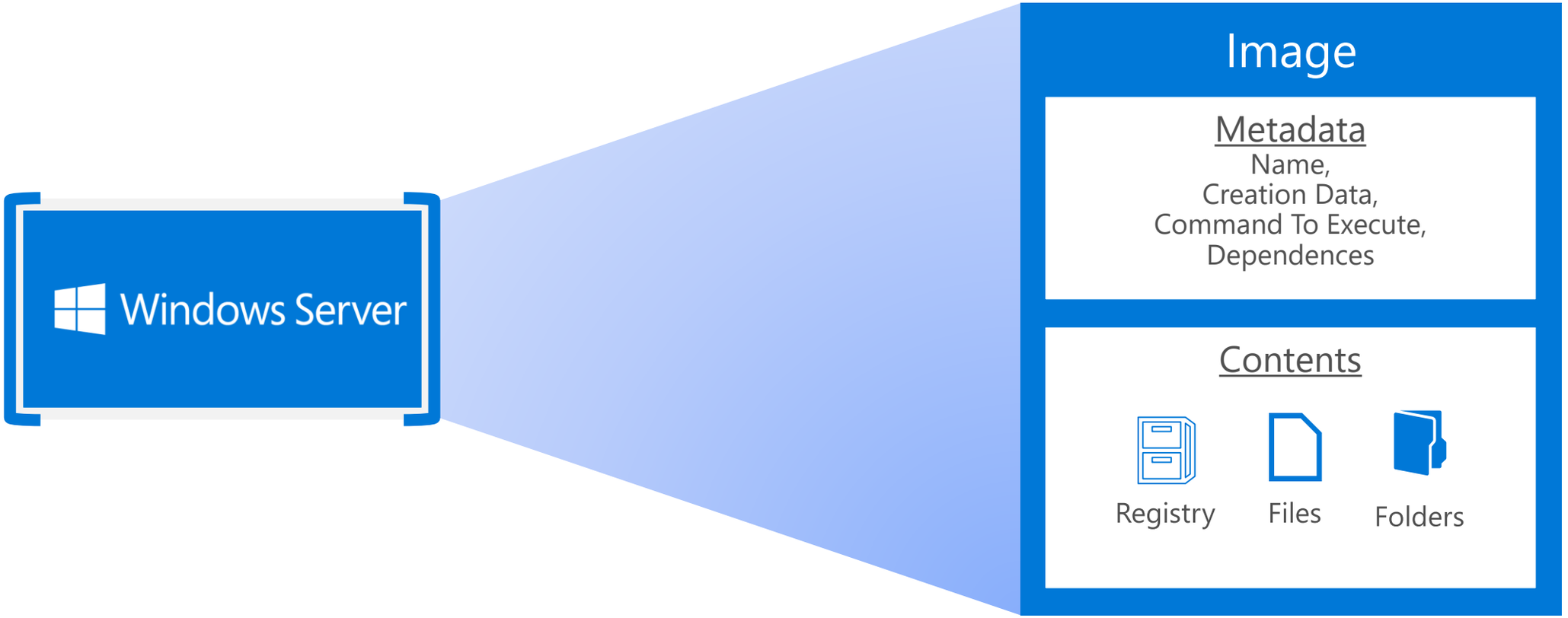
- Name (network name and management name)
- Interactive or Service
- Network configuration
- Resource management
- Volume mappings
- Isolation level



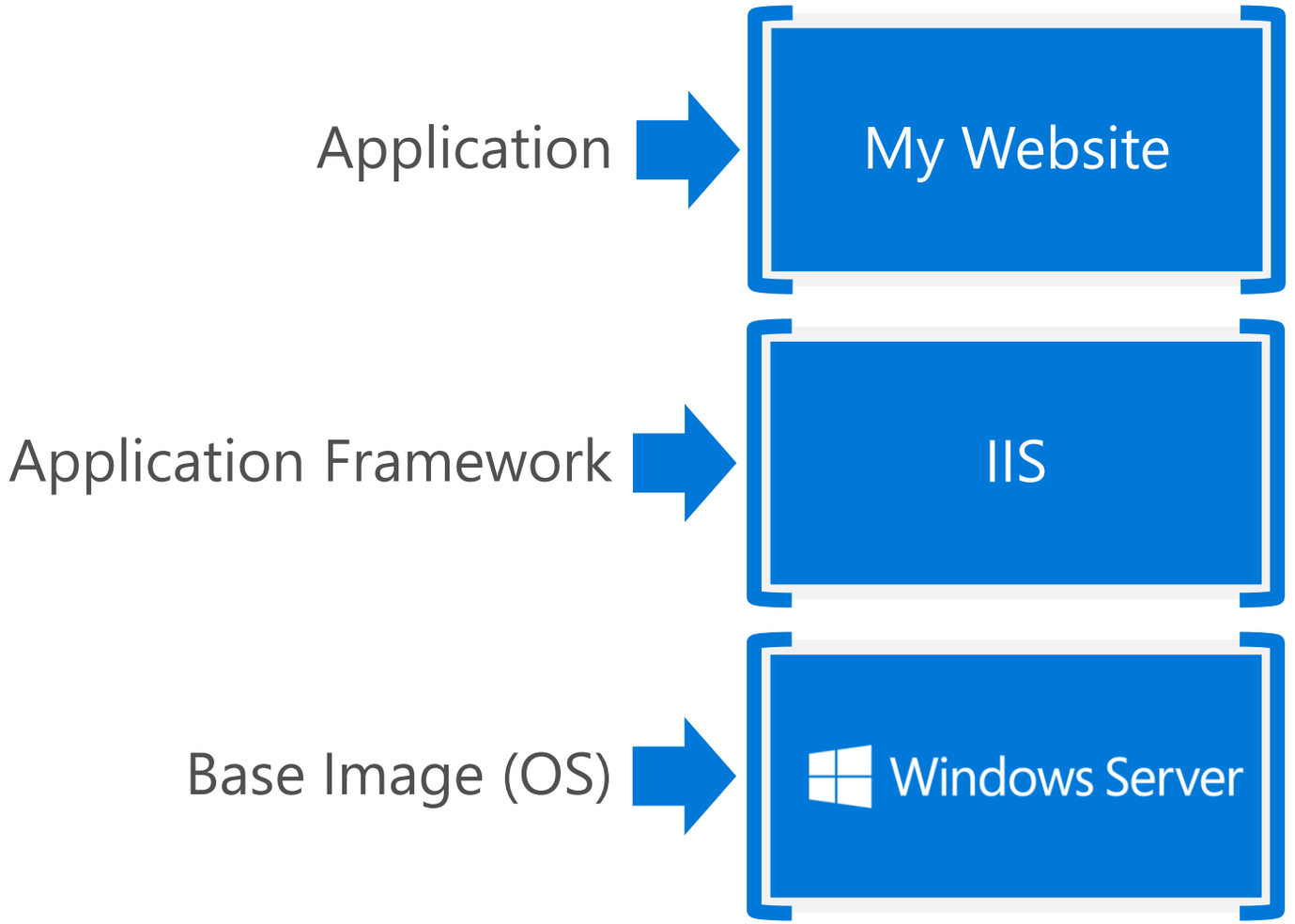
What is a container image?

Container Image

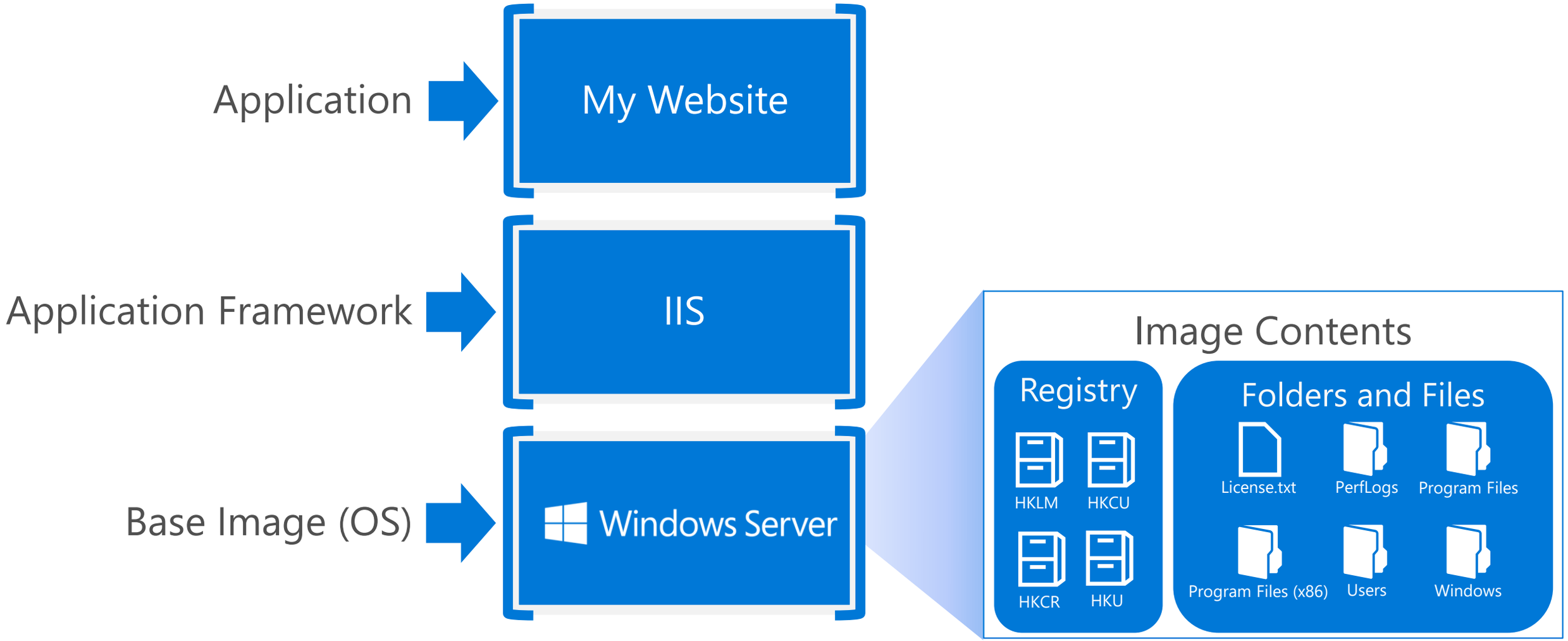
Analogous to a VHD and config file to a virtual machine
Created by running a container and capturing changes
Changes include files and registry



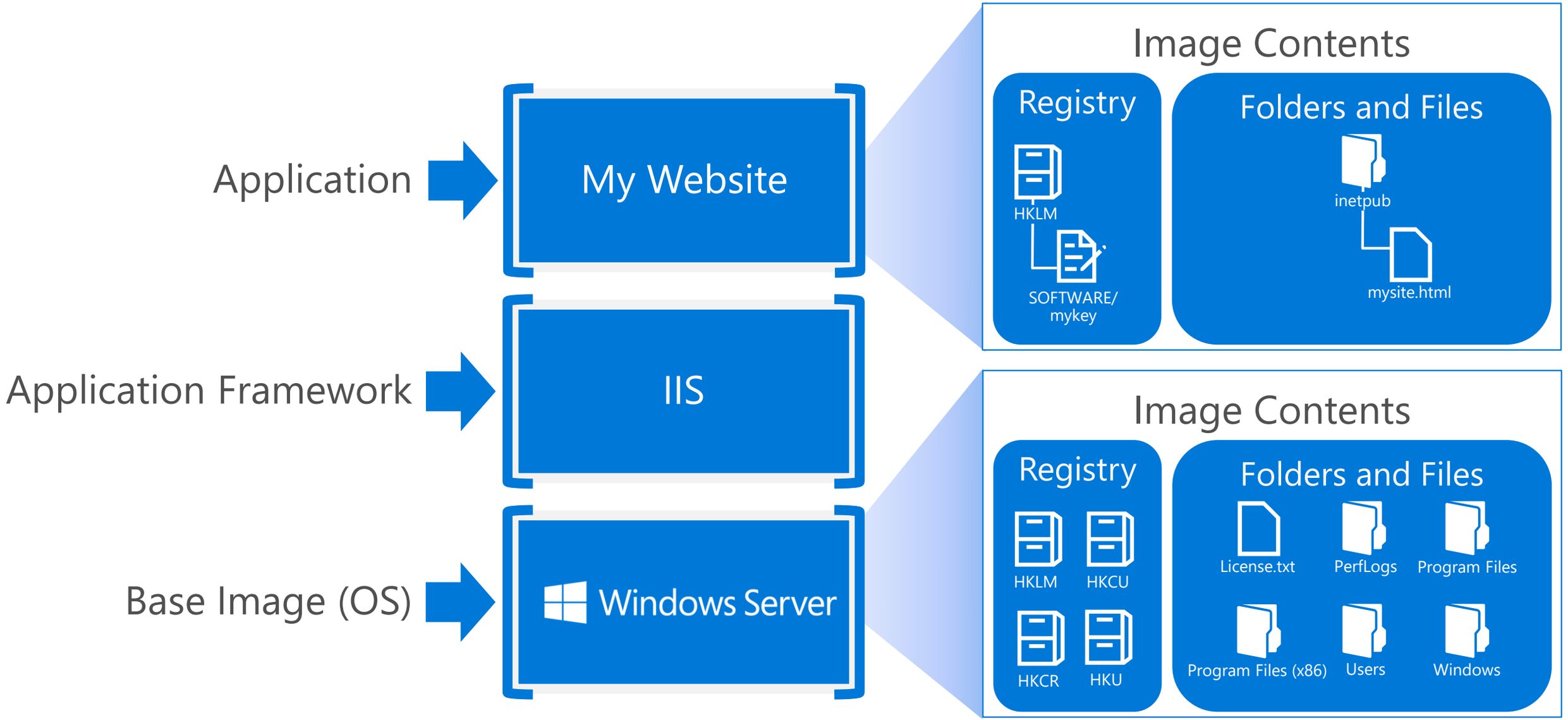
Container Images



Container Images



Container Images



Automated Image Building

Docker Build and Dockerfiles

Method for automated container image build
Consumed when running “docker build”
Caches unchanged commands
Integrates into Docker Hub

Examples

IIS

```
FROM windowsservercore
RUN powershell –command Add-WindowsFeature Web-Server
```

Website

```
FROM iis
ADD mysite.htm inetpub\mysite.htm
```

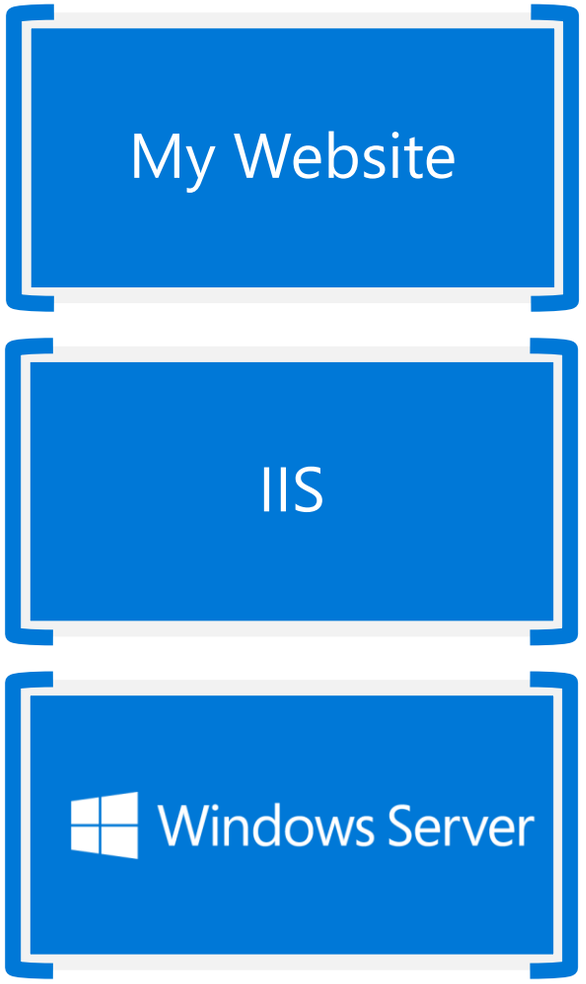


Image Registries

What is a registry?

Stores container images

- Images are **Pushed** into a registry
- Images are **Pulled** from a registry
- Images are **Searched** for within a registry

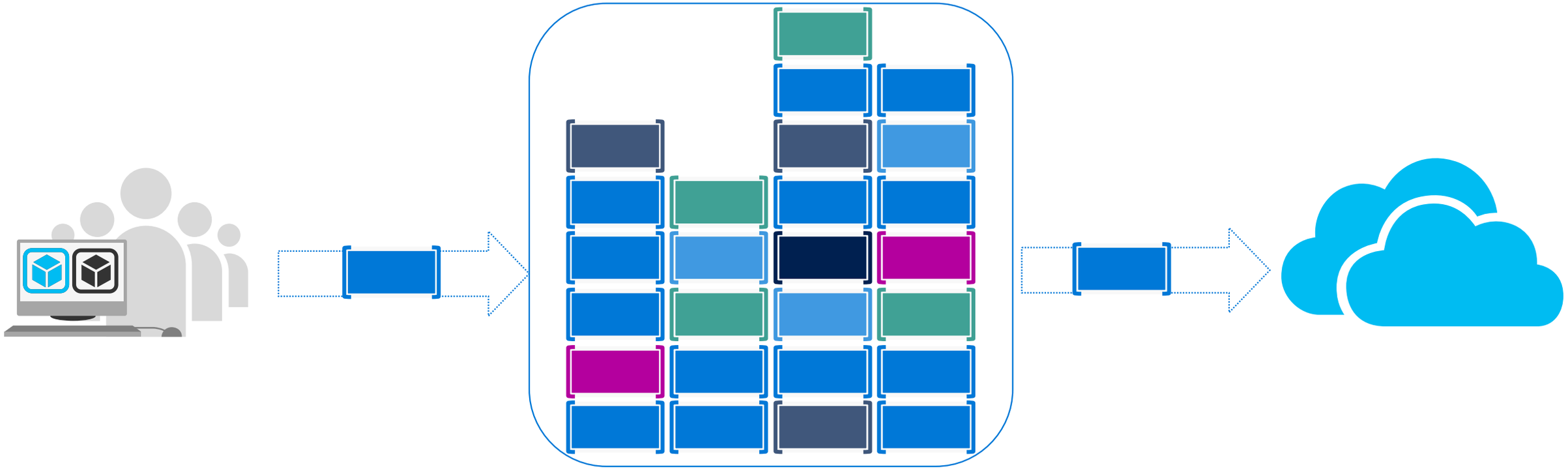


Image Registries

Docker Hub and Docker Store

Public, Official and Private image repositories
Granular access controls with organization support
Automated image build support

Docker Trusted Registry

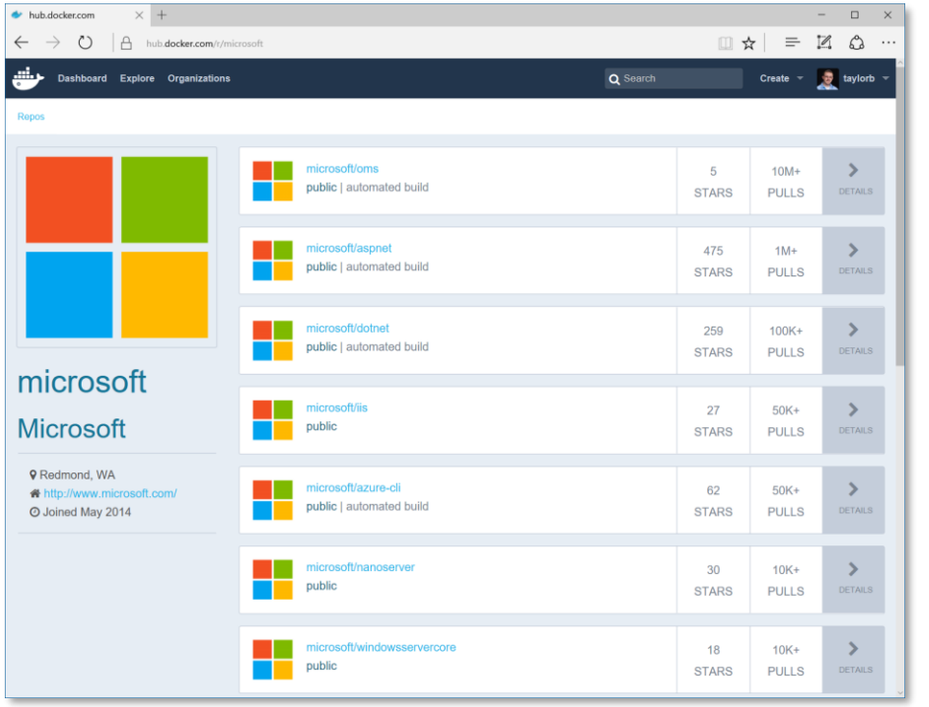
Enterprise Grade Private Registries
Runs on your infrastructure (on-prem or cloud)
Active Directory and Role Based Access Controls

Azure Container Registry

Store and manage container images across Azure deployments
Maintain Windows and Linux container images
Same API and Tools as Docker Hub/Store/Registry

Docker Registry

Open source foundation of Hub and DTR
Runs on your infrastructure (on-prem or cloud) as a container
https://docs.docker.com/registry and or https://github.com/docker/distribution



What about…

Licensing

Windows Server 2016 feature differentiation and core-based pricing		
Feature	Datacenter	Standard
Core functionality of Windows Server	•	•
OSEs / Hyper-V containers	Unlimited	2
Windows Server containers	Unlimited	Unlimited
Host Guardian Service	•	•
Nano Server*	•	•
Storage features including Storage Spaces Direct and Storage Replica	•	
Shielded Virtual Machines	•	
Networking stack	•	
Core-based pricing**	$6,155	$882

*Software Assurance is required to deploy and operate Nano Server in production.

**Pricing for Open (NL) ERP license for 16 core licenses. Actual customer prices may vary.

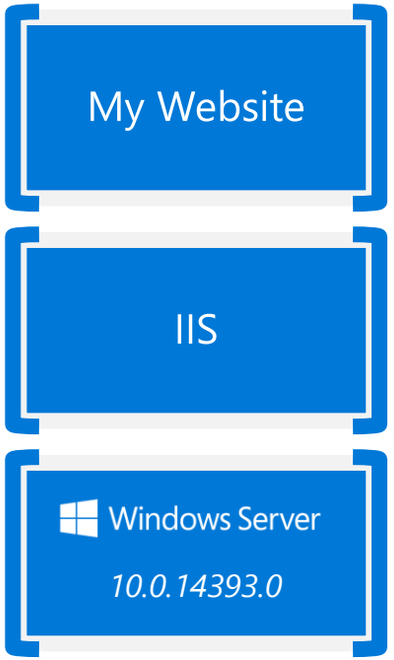
https://www.microsoft.com/en-us/cloud-platform/windows-server-pricing

Patching and Updates

Update Container OS Image

Pull updated base image

Rebuild containers using dockerfiles

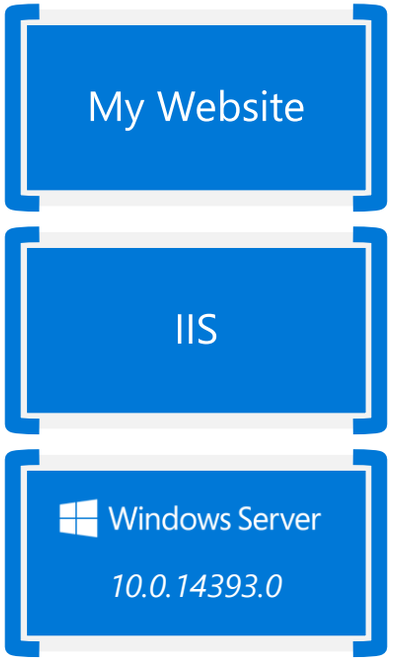


Patching and Updates

Update Container OS Image

Pull updated base image

Rebuild containers using dockerfiles

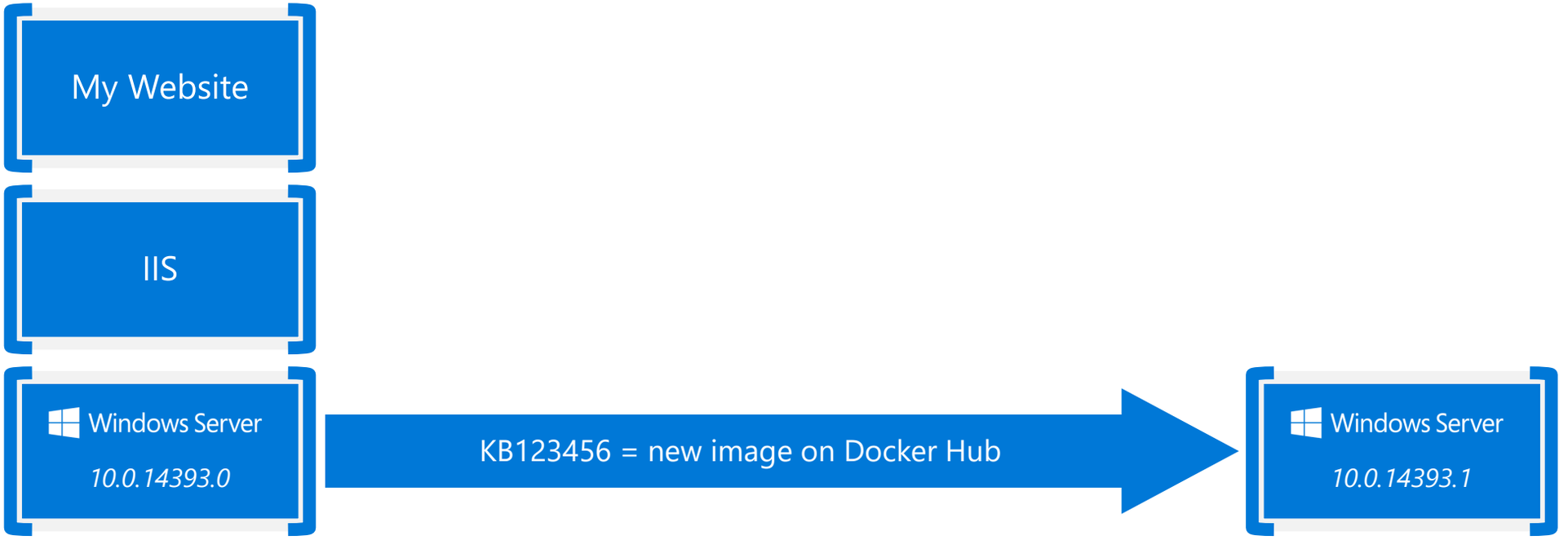


Patching and Updates

Update Container OS Image

Pull updated base image

Rebuild containers using dockerfiles

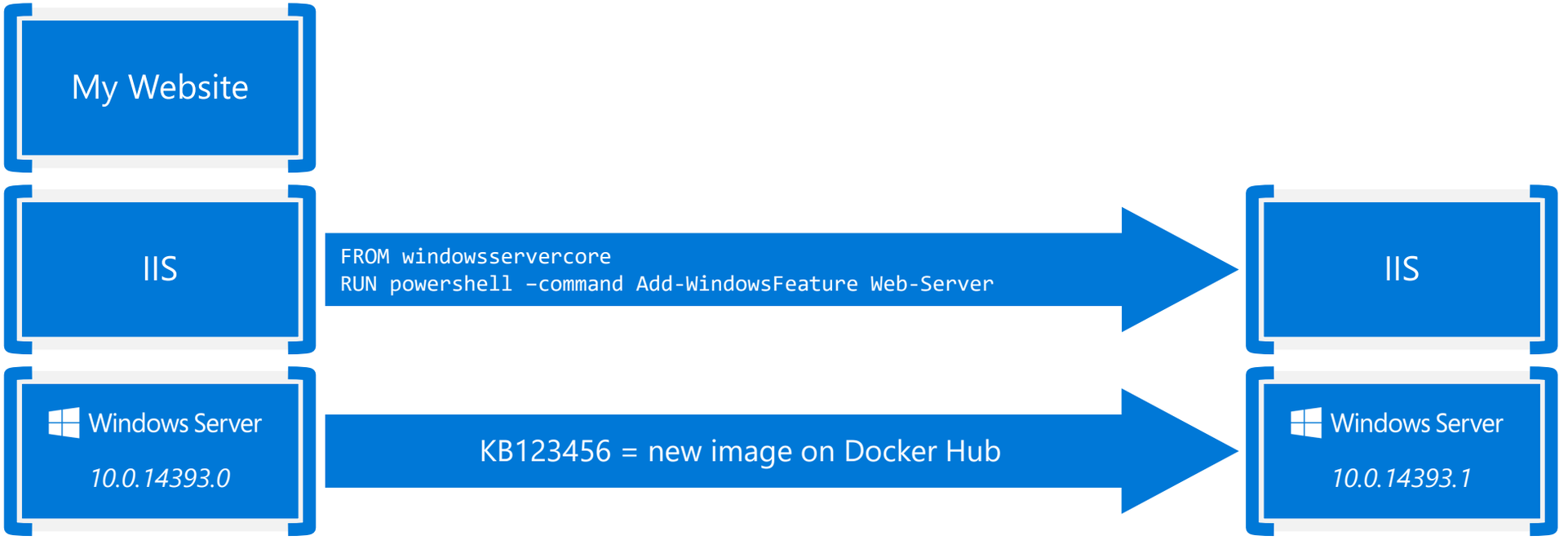


Patching and Updates

Update Container OS Image

Pull updated base image

Rebuild containers using dockerfiles

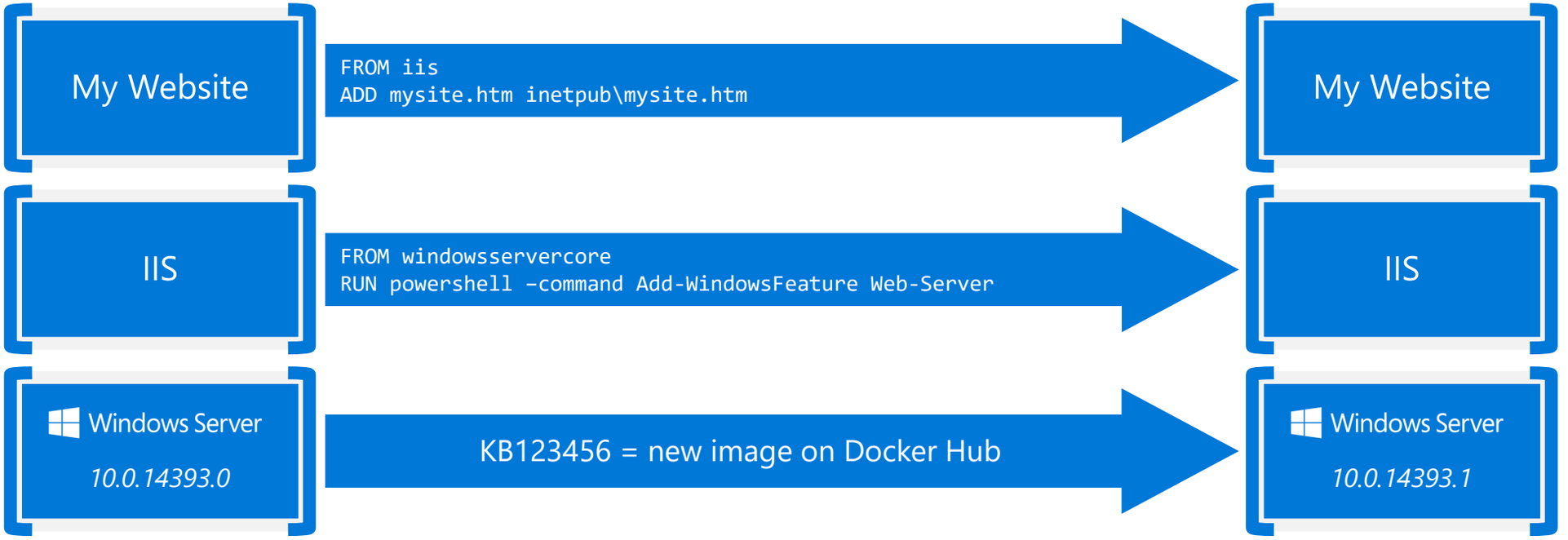


Patching and Updates

Update Container OS Image

Pull updated base image

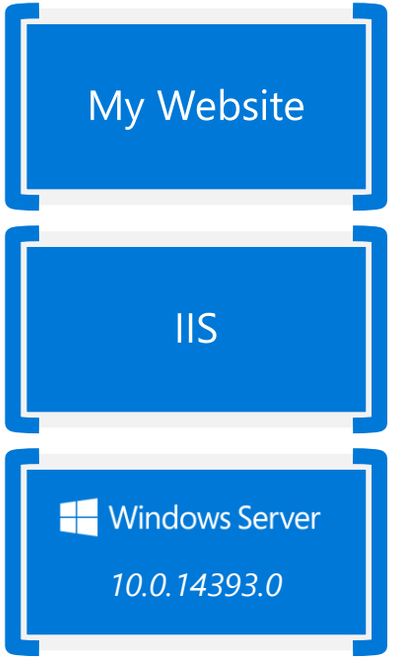
Rebuild containers using dockerfiles



Patching and Updates

Update as a new layer

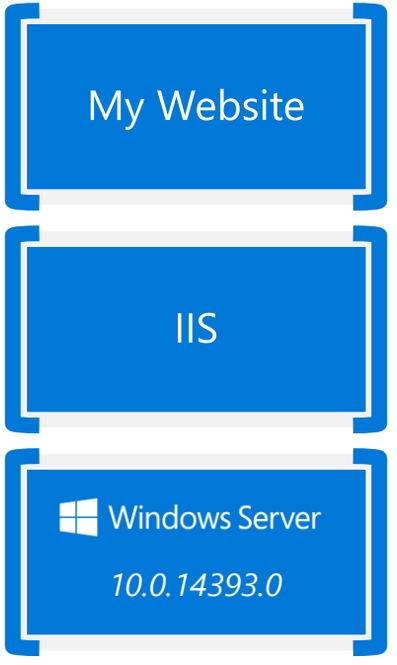
Download update in container (ala run Windows Update in the container)
When container is stopped update is applied as a new layer



Patching and Updates

Update as a new layer

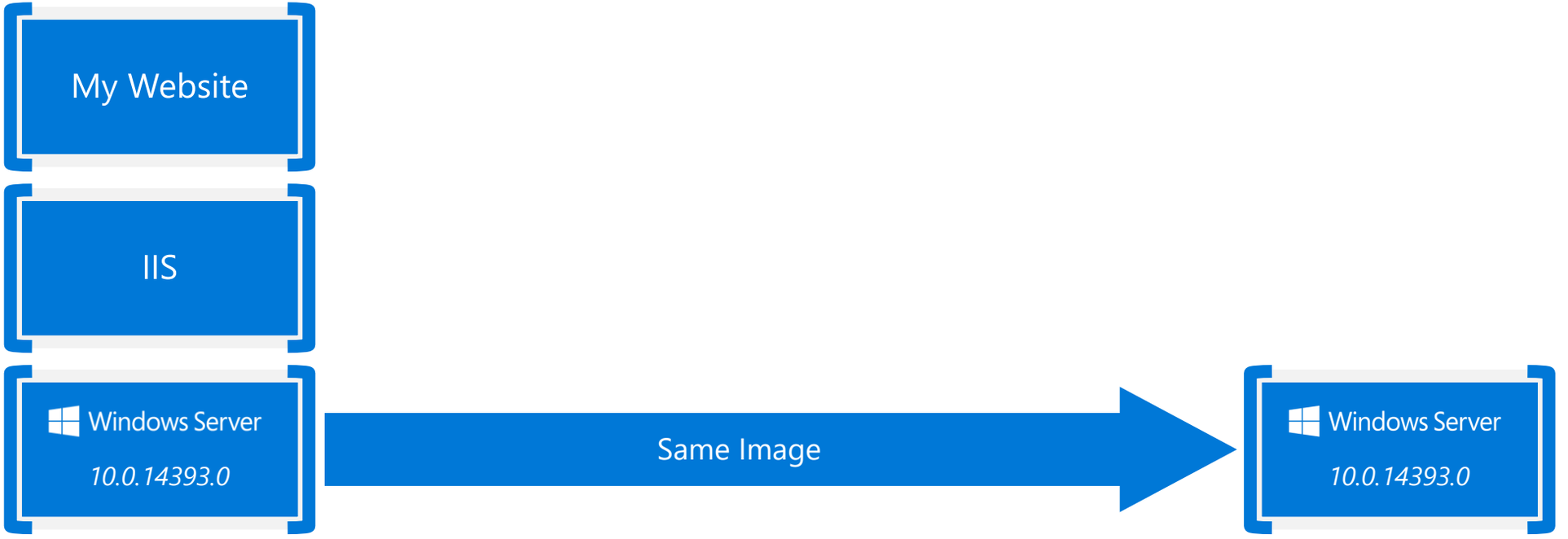
Download update in container (ala run Windows Update in the container)
When container is stopped update is applied as a new layer



Patching and Updates

Update as a new layer

Download update in container (ala run Windows Update in the container)
When container is stopped update is applied as a new layer



Patching and Updates

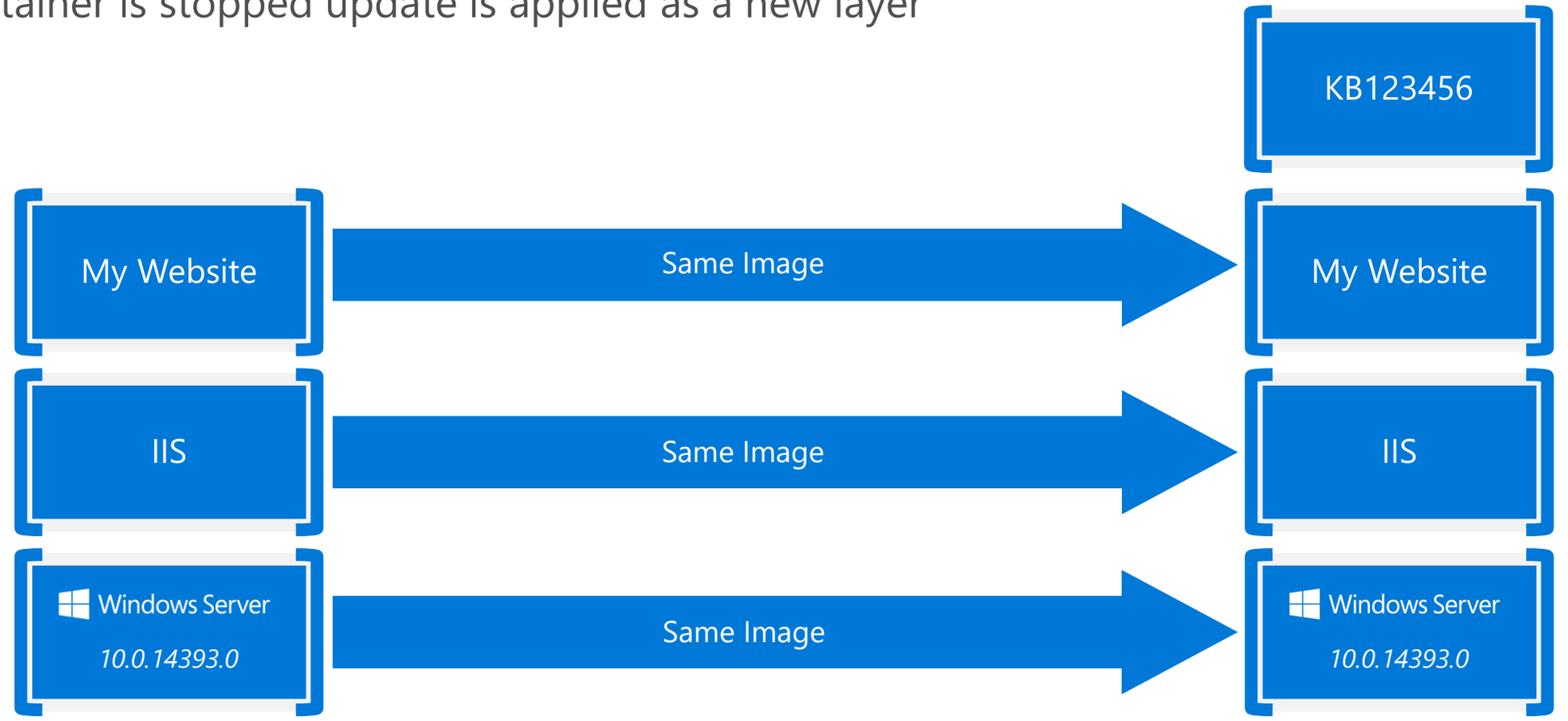
Update as a new layer

Download update in container (ala run Windows Update in the container)
When container is stopped update is applied as a new layer

Patching and Updates

Update as a new layer

Download update in container (ala run Windows Update in the container)
When container is stopped update is applied as a new layer



Resource Controls

CPU

Percent of host CPU container can consume

Memory

Maximum memory container can consume

Disk

Maximum IO bandwidth on the system drive
Maximum IOPs limit on the system drive

Network

Platform support for egress caps

Networking

Fully Manageable with Docker

Network creation/enumeration
Service Discovery

Optimized for Microsoft Cloud Stack

Advanced network policy (ACLs, QoS) can be assigned per container endpoint
Load Balancing can be handled through the Microsoft Software Load Balancer (Coming Soon)

Area of Regular and Continuous Innovation

Docker tooling support (Compose) for networking (Limited Support at GA)
Integration with other Orchestrators (Kubernetes, Swarm, etc.)
Native Overlay network driver
Multiple networks (NAT and overlay) per host

Storage

Container Image

Not designed for persistent data
Not designed for secrets

“Volumes”

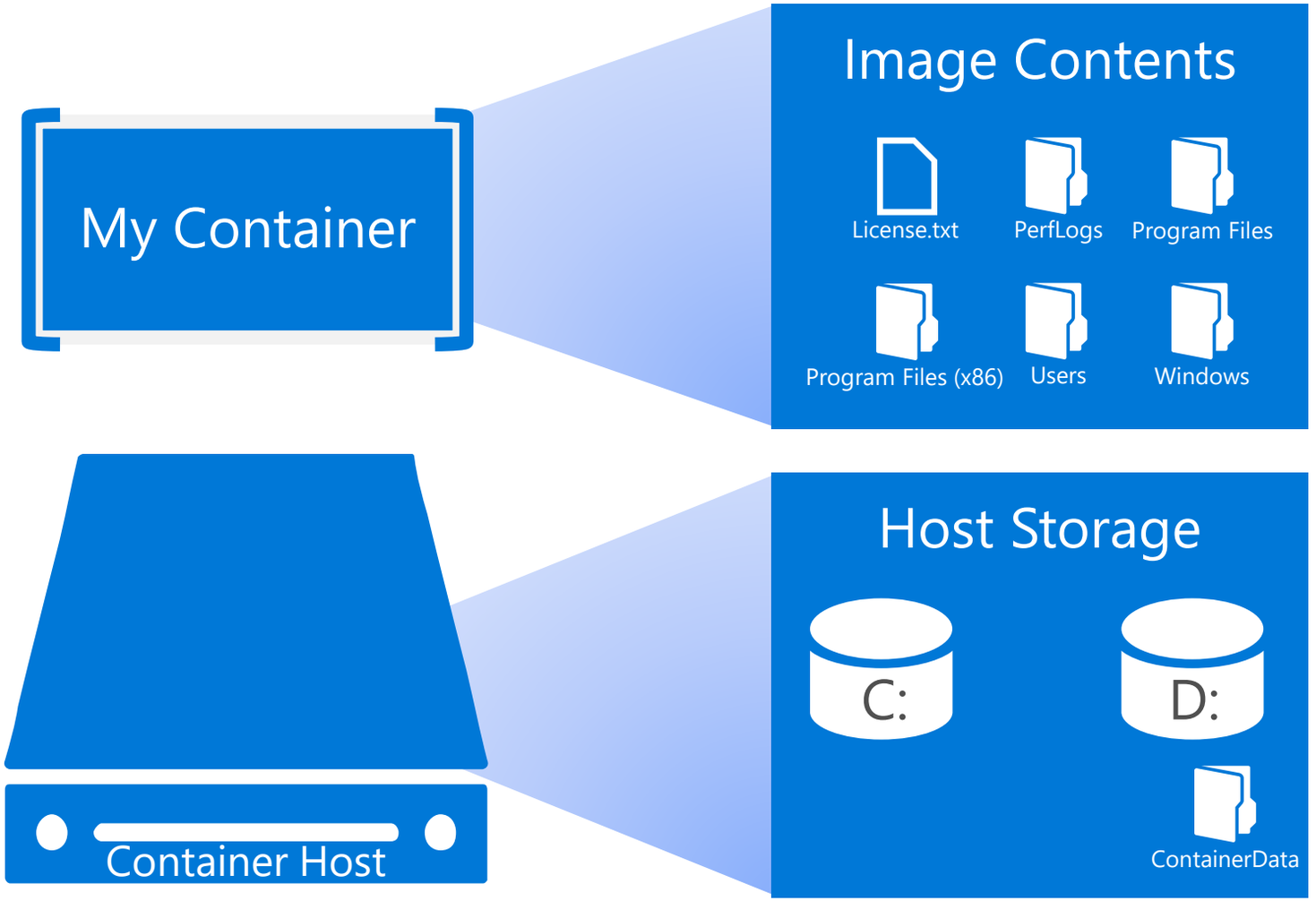
Enables storage persistence
Enables mapping of storage into containers
Read-Only or Read/Write
Multiple containers on the same host can access the same location
Plug-In Architecture

Network Storage

Containers access SMB shares
Accessed though the containers network

Volume Mapping

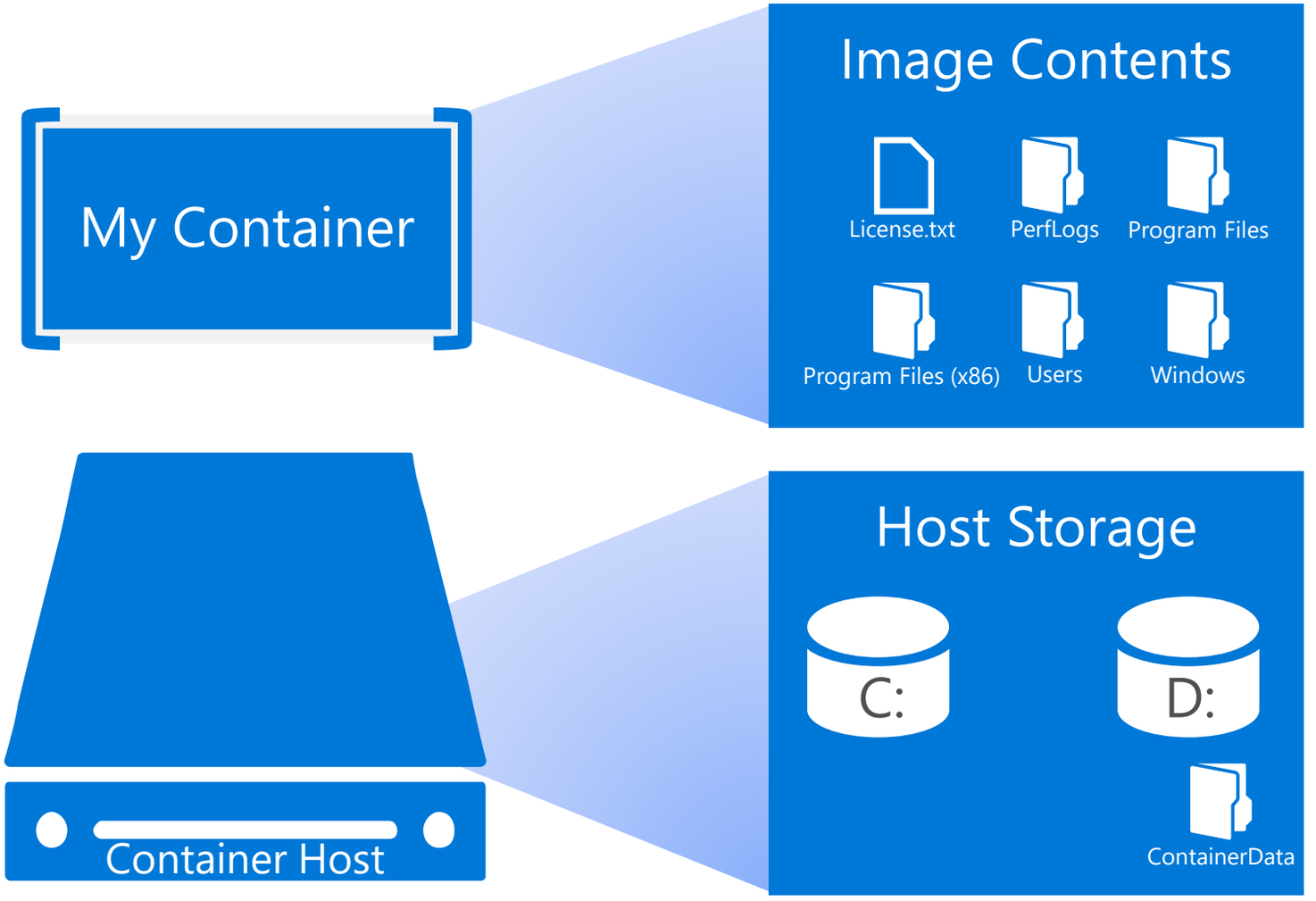
Volume Mapping



Storage Volume Mapping Example

Running a Container

```
docker run
-v d:\ContainerData:c:\data
mycontiner
```

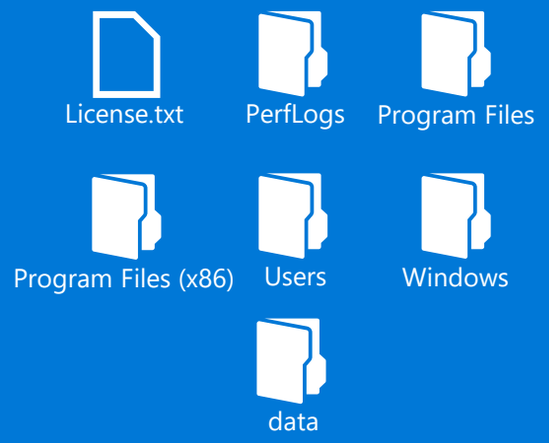


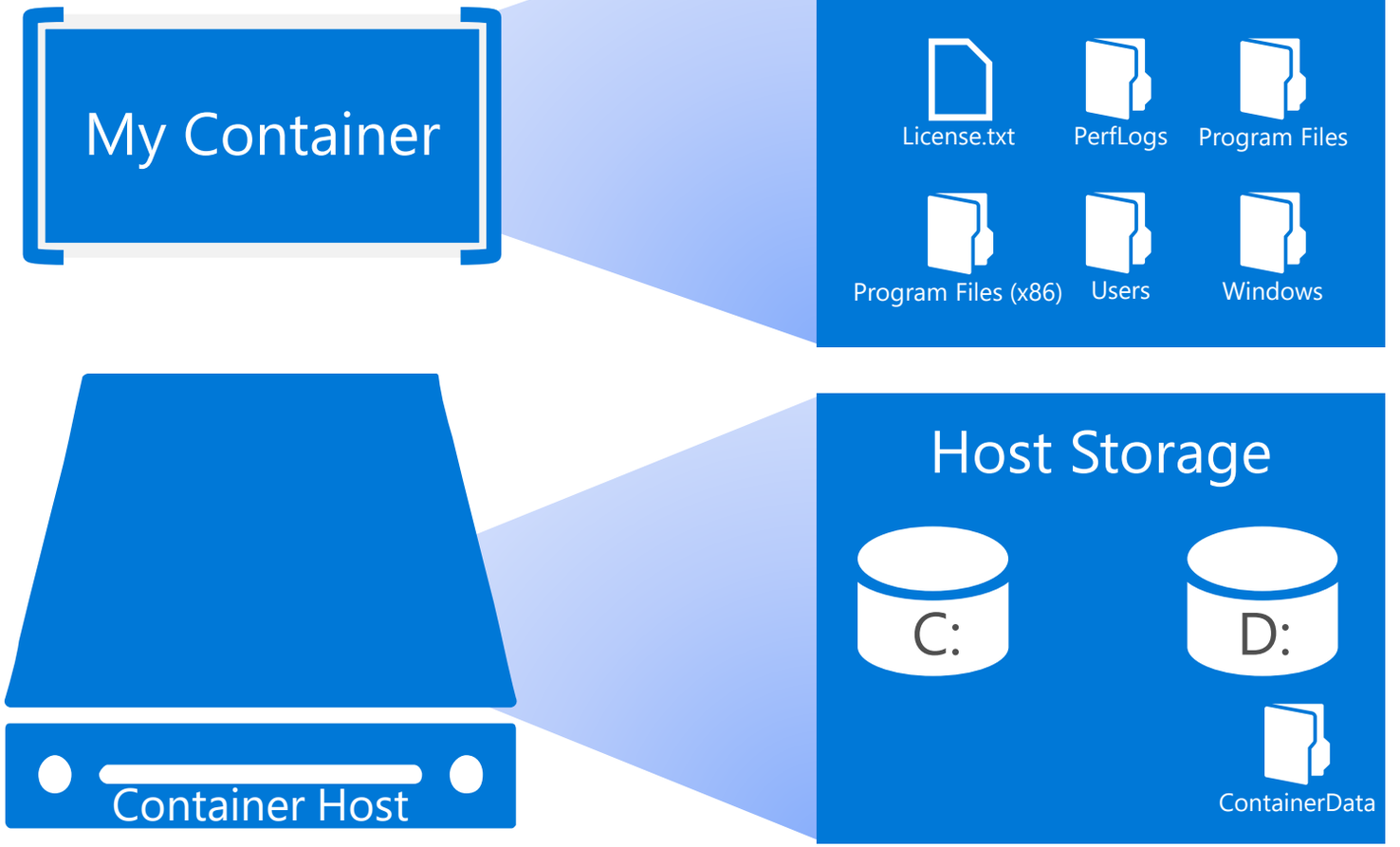
Volume Mapping

Running a Container

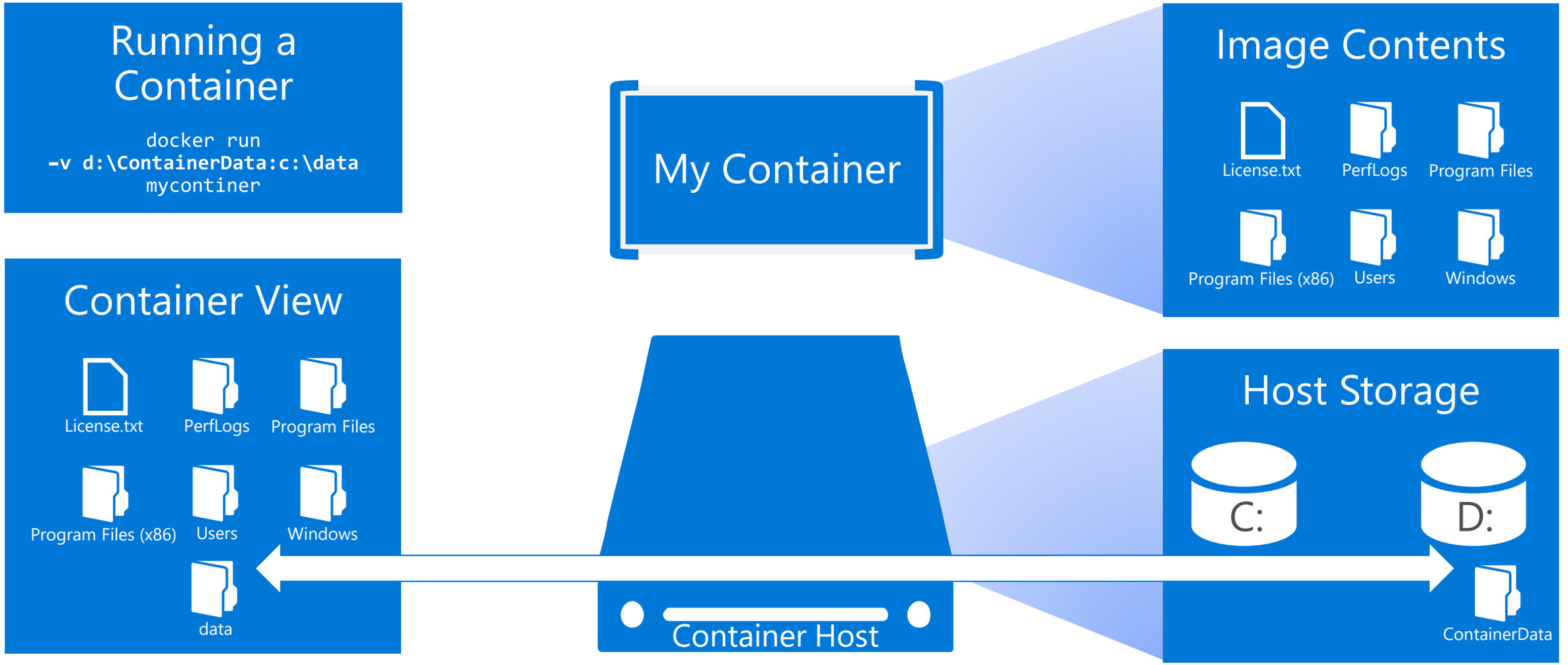
```
docker run
-v d:\ContainerData:c:\data
mycontiner
```

Container View





Volume Mapping



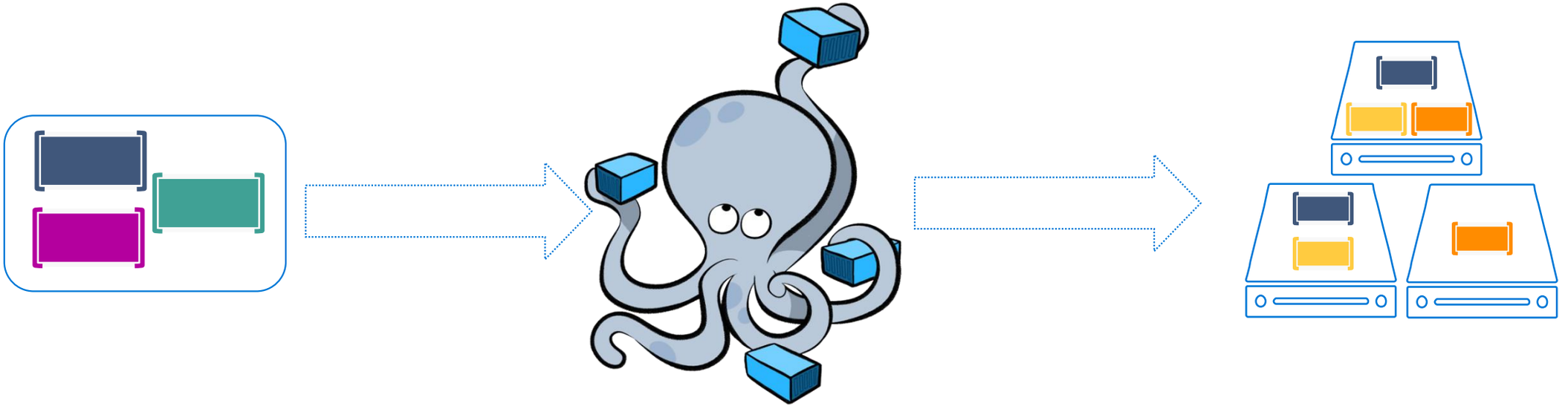
Automation and Management

Composition and Orchestration

Applications typically comprised of multiple containers

Containers typically hosted across a cluster of nodes

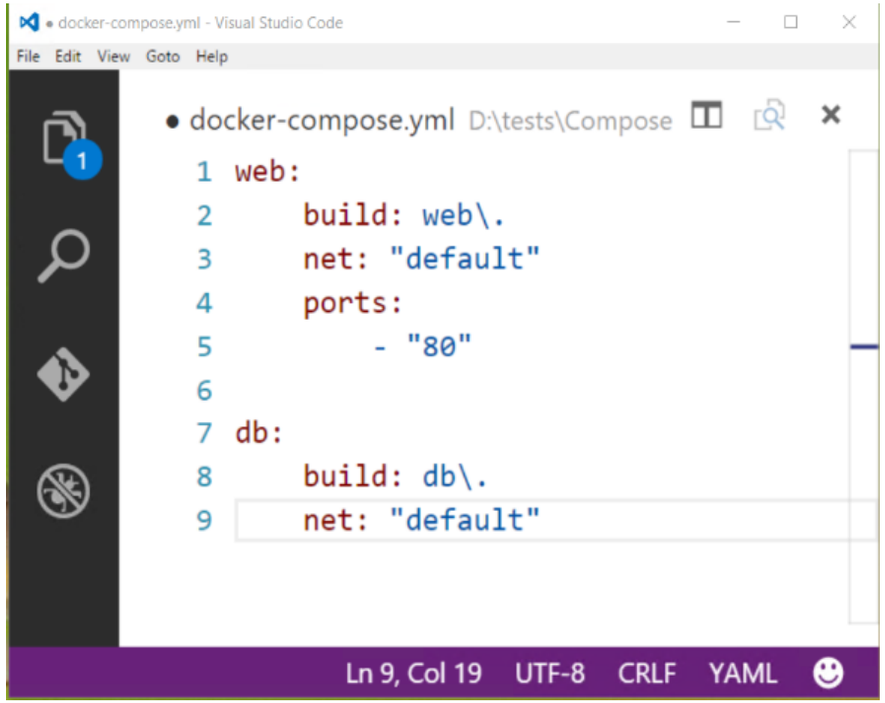
Orchestration tooling automates this



Composition and Orchestration

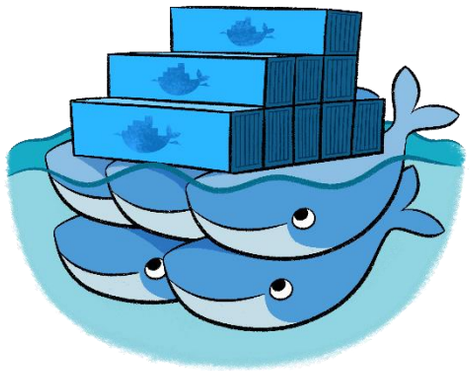
Docker Compose

Define application as separate containers
Manage different containers as a unit
Scale parts of application as needed



Docker Swarm

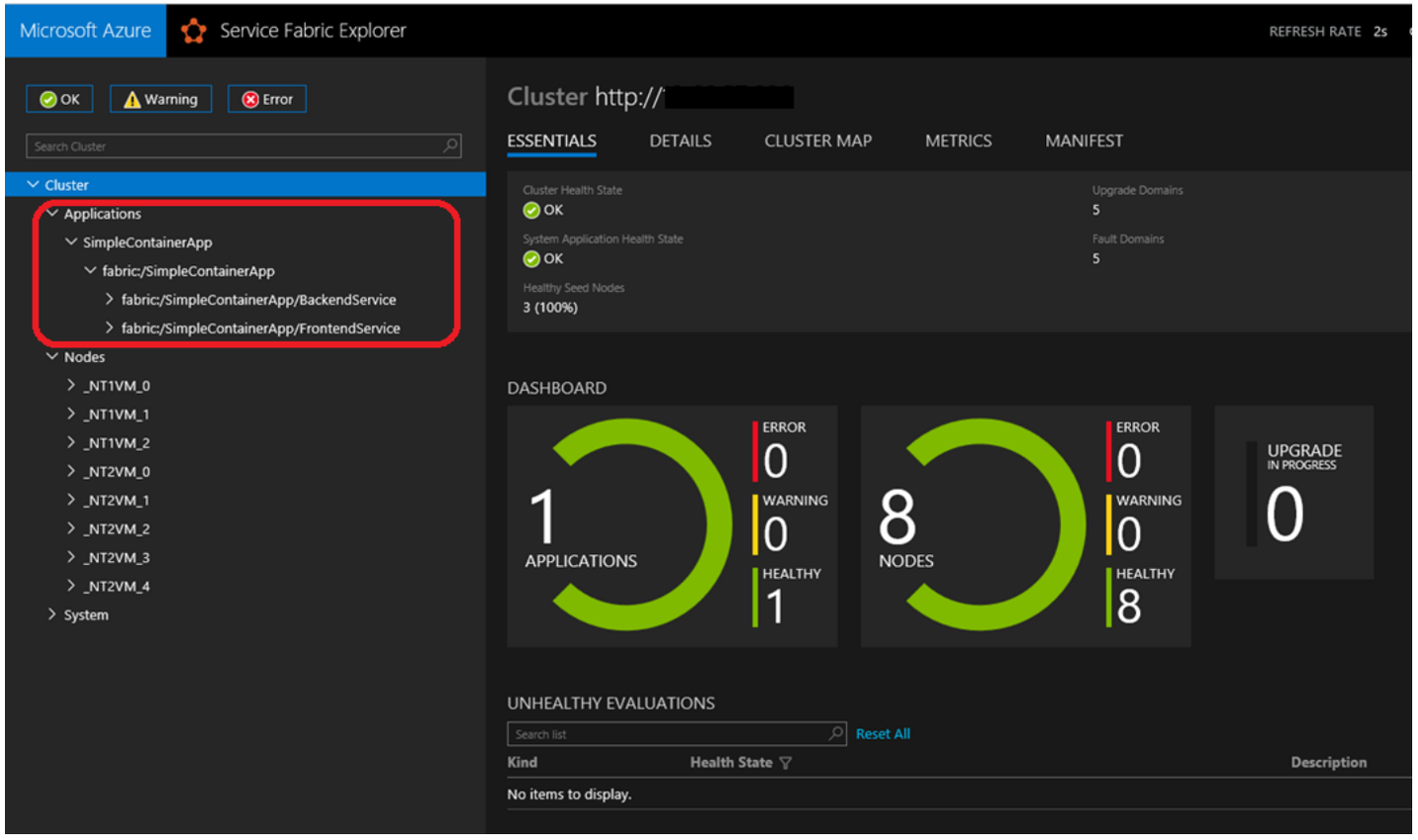
Aggregate container hosts
Supports tagging, affinity/anti-affinity



Composition and Orchestration

Azure Service Fabric

Microservice and orchestration platform
Build applications as containers **and/or** microservices
Available on Windows & Linux
Built-in cross-container communication
Web based management UI
Available On-Prem, Azure or other Clouds

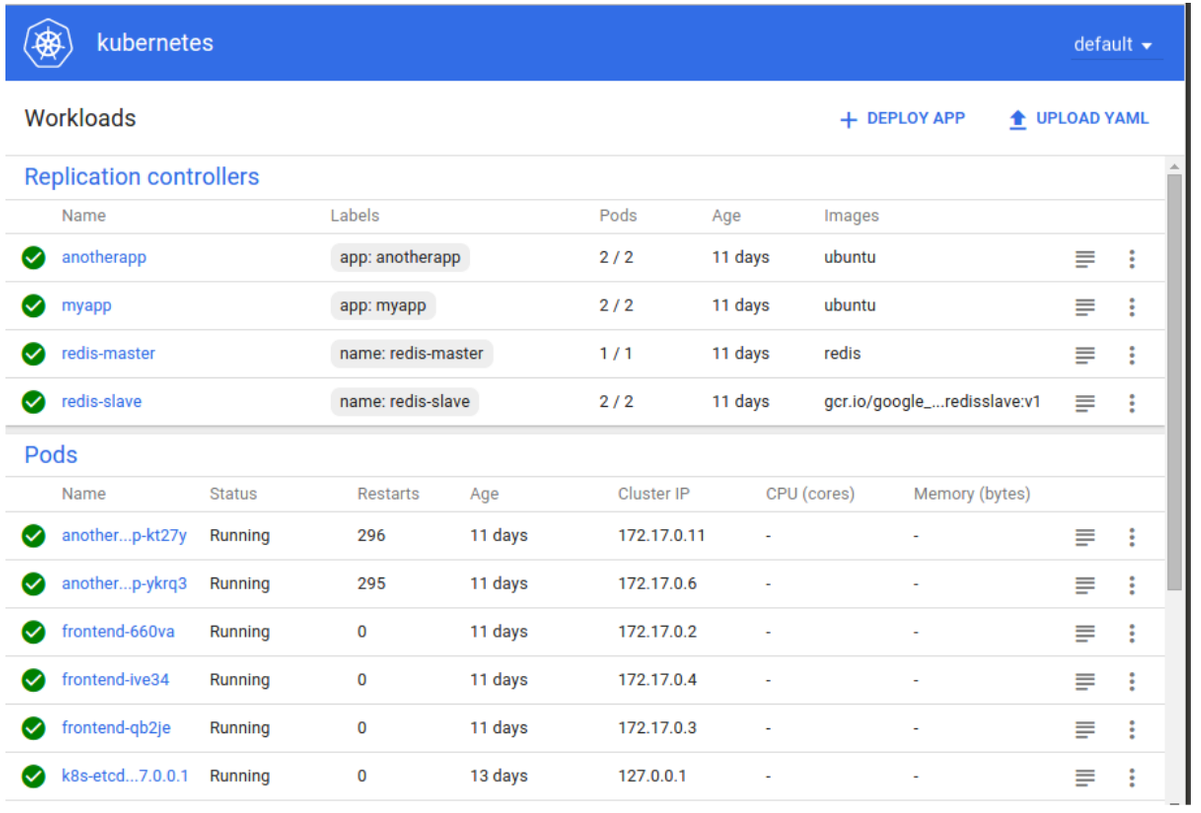


Composition and Orchestration

Kubernetes

Open source project started by Google

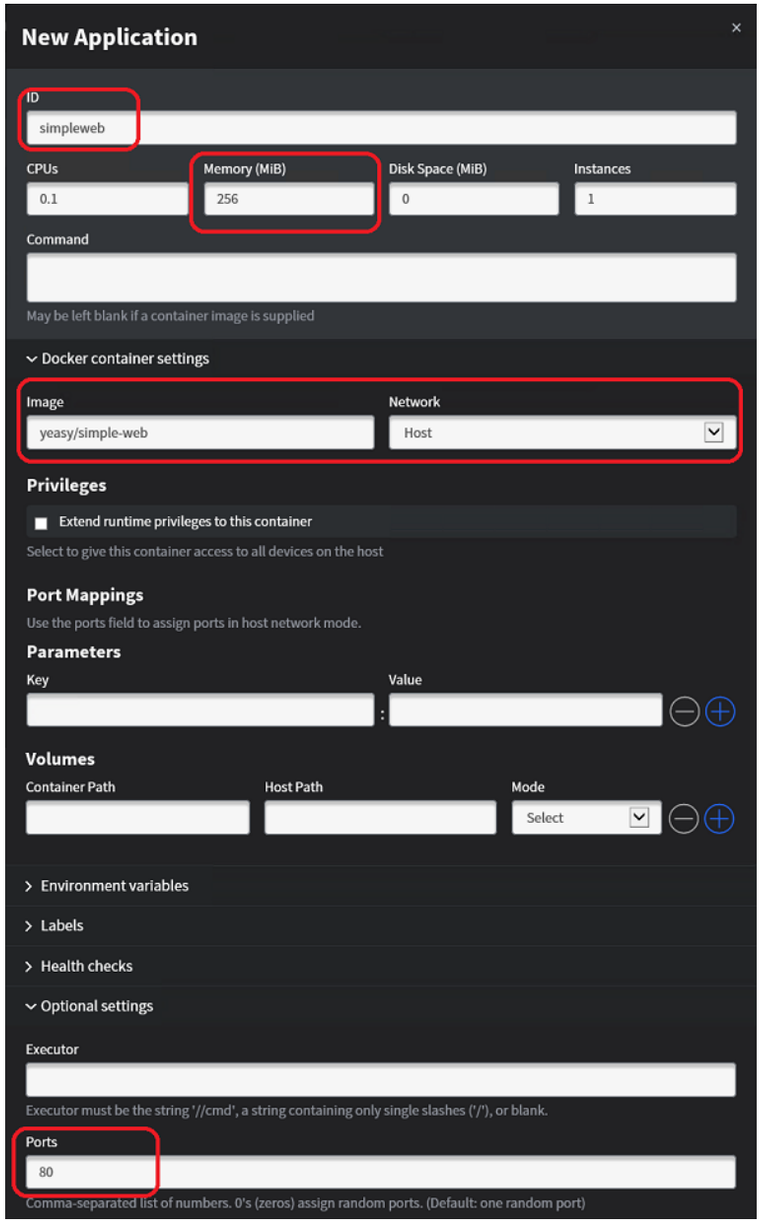
Windows support being added though community partnership spear headed by Apprenda



Composition and Orchestration

Mesos/Mesosphere + Marathon

Aggregates container hosts
Web based UI
Service Launch and Discovery



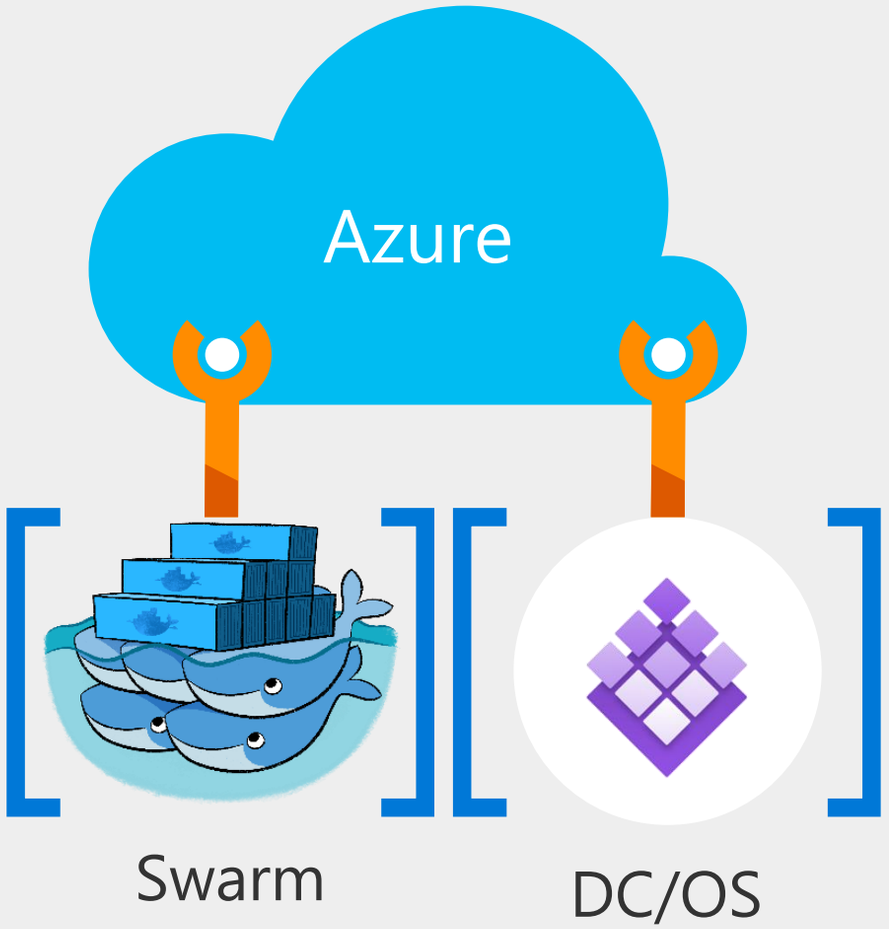
Azure Container Service

Standard Docker tooling and API support

Streamlined provisioning of Docker Swarm and DCOS

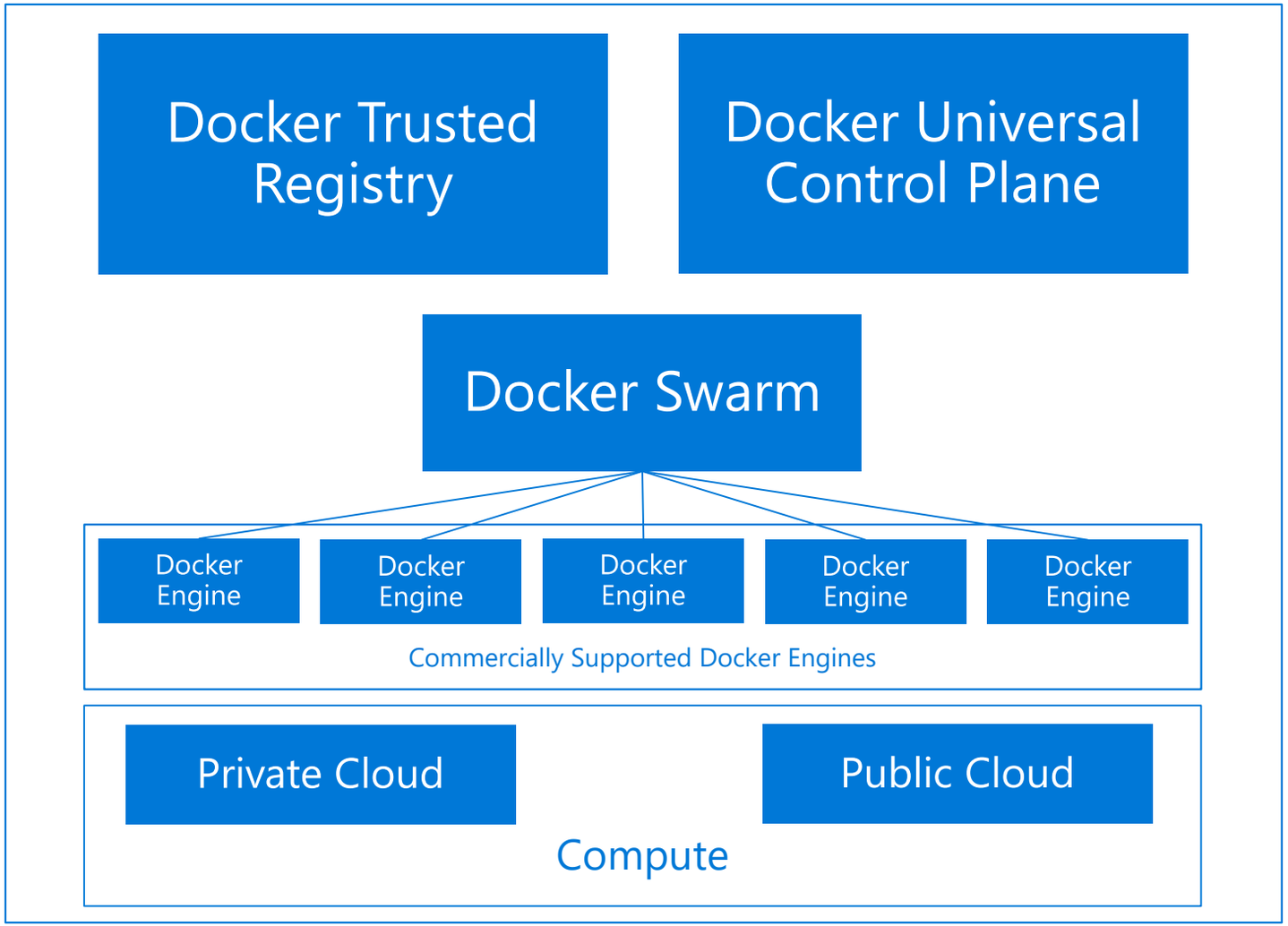
Linux and Windows Server containers

Azure and Azure Stack



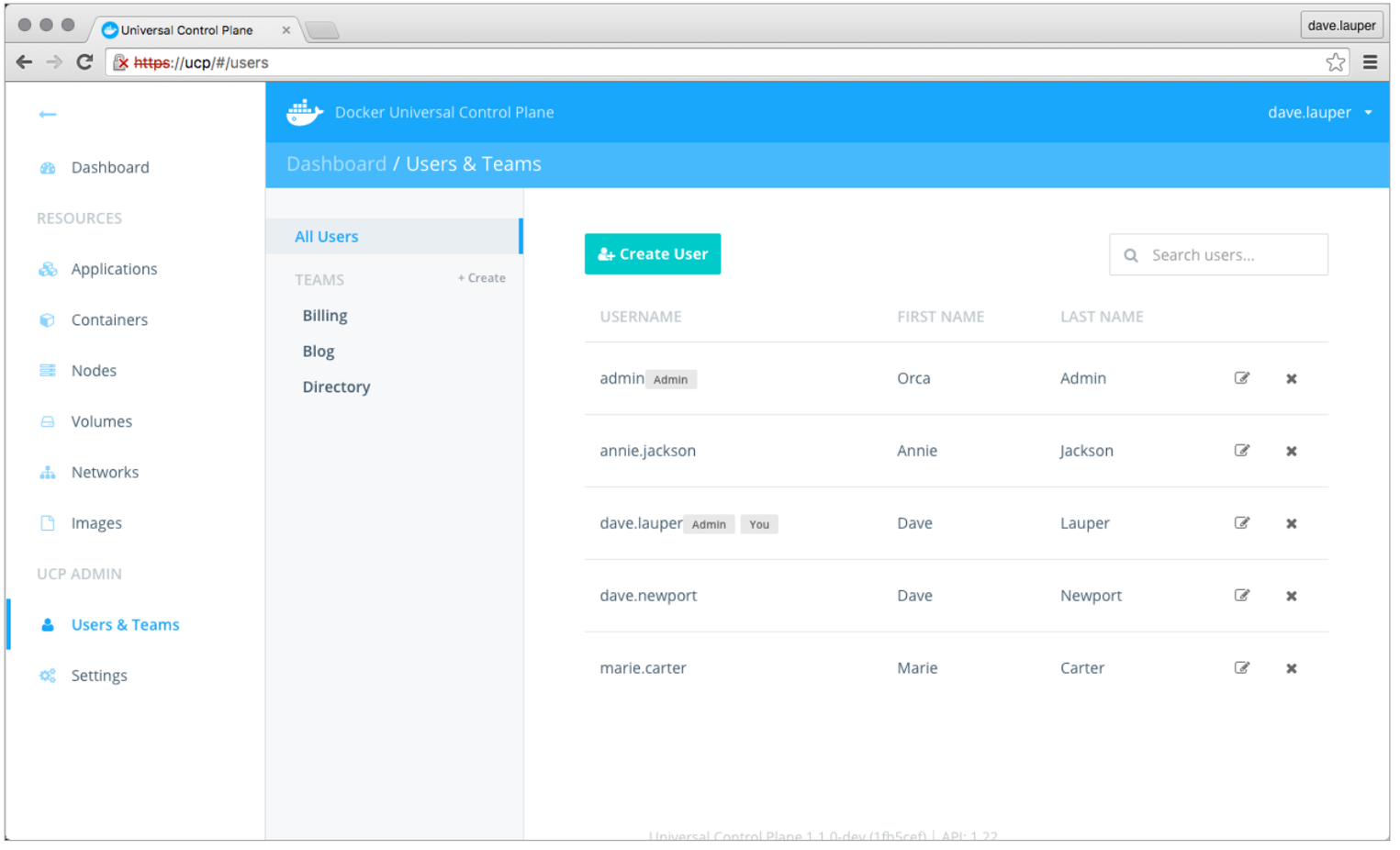
Management and Monitoring Tools

Docker Datacenter



Management and Monitoring Tools

Docker Datacenter



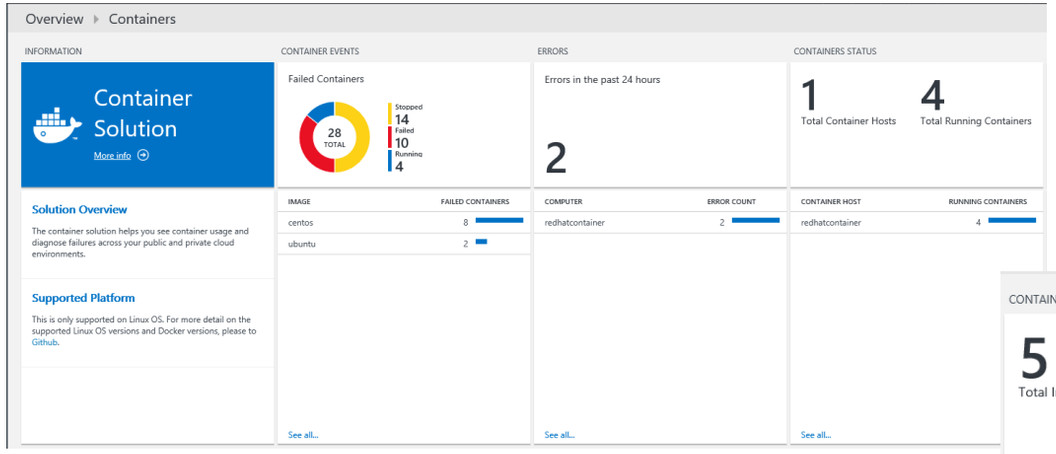
Microsoft Operations Management Suite

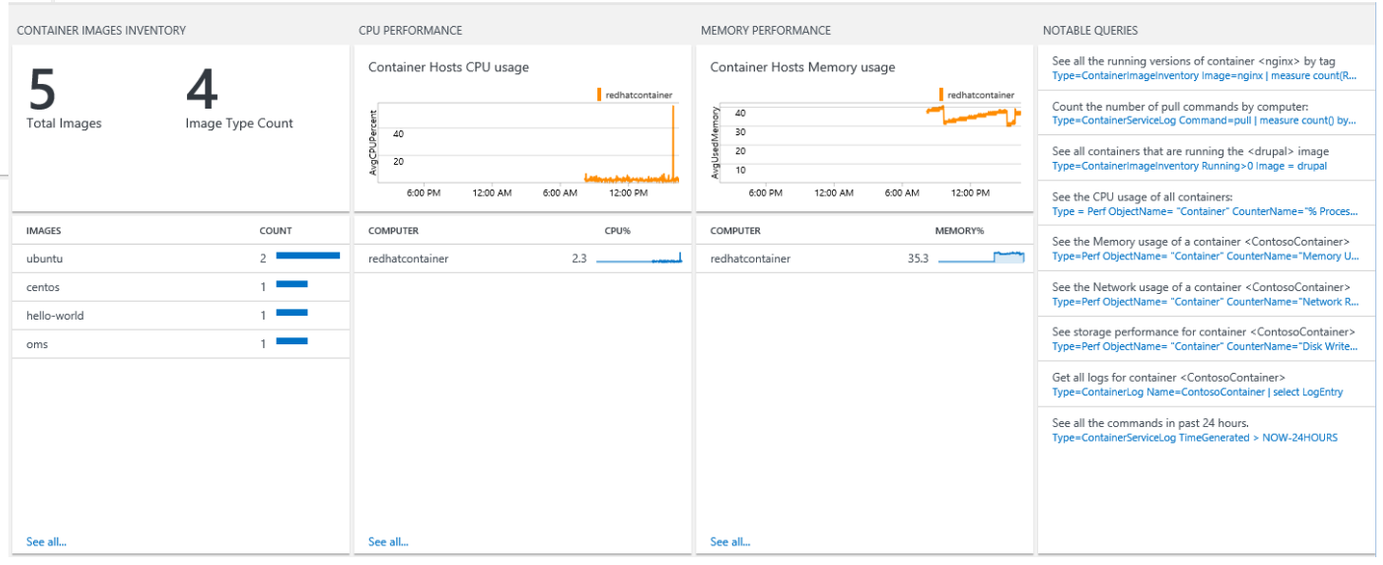
Monitoring Solution for Docker

Syslog events

Performance metrics

Container data

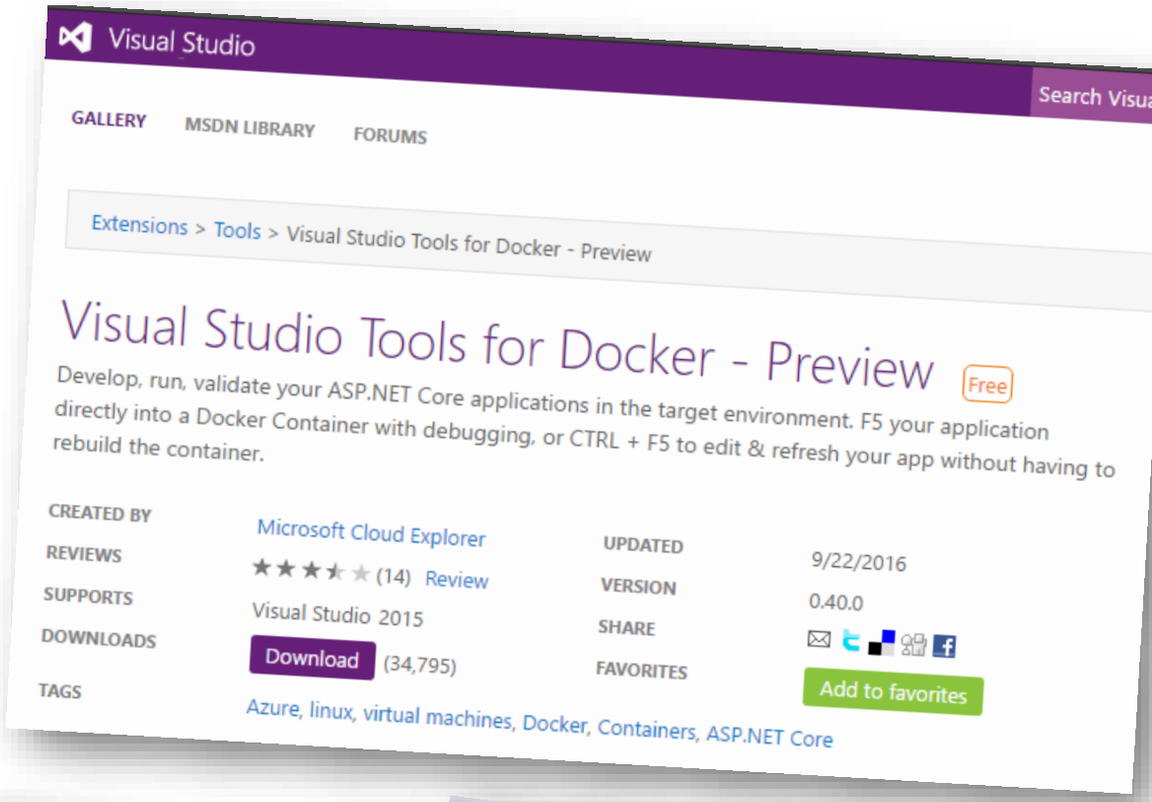


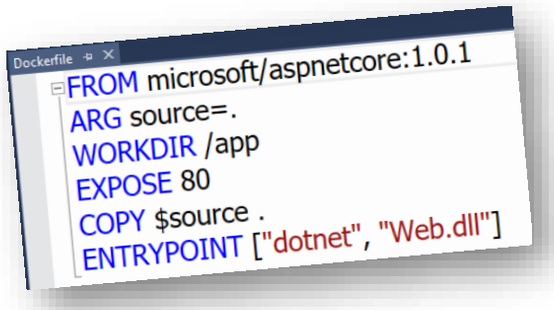


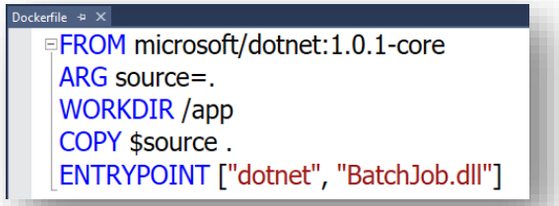
Development Tools

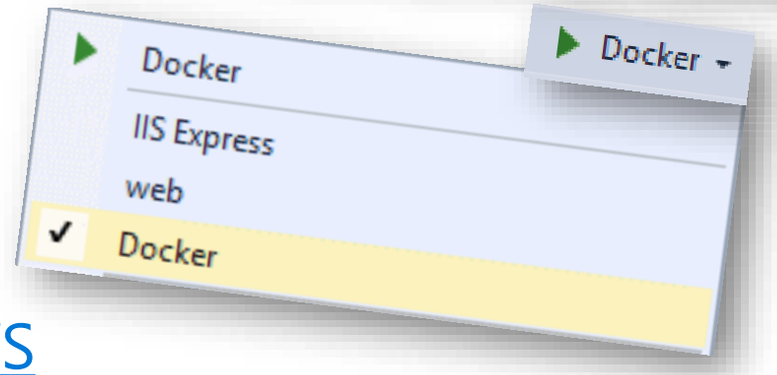
Visual Studio Docker Tools

- Run, Debug, Test Web & Console apps in docker containers
  - *Linux today, Windows Server & Nano Server coming soon*
- F5 Debugging
- Edit & Refresh of code
- Scaffolds docker assets
  - Dockerfile, docker-compose.yml









aka.ms/DockerToolsForVS

NEWS

Windows Server 2016 Getting Linux Containers

By Scott Bekker ■ 04/20/2017

Microsoft this week announced that a new type of Linux container that runs on Windows Server 2016 will be coming soon.

The move, which was unveiled at this week's DockerCon 2017 in Austin, Texas, would break down a fundamental wall in deployment scenarios for containers to date. For now, Linux containers can only run on Linux host operating systems and Windows containers can only run on Windows

Most Popular Articles

- OpenSSH Windows Nearing Completion
- Does Everyone Still Need Windows?
- Windows 10 Creators Update Tools and Documentation Released
- Microsoft Releases Windows 10 Creators Update and Security Patches
- A Brave New World: Top Features Coming to SQL Server 2017

Microsoft