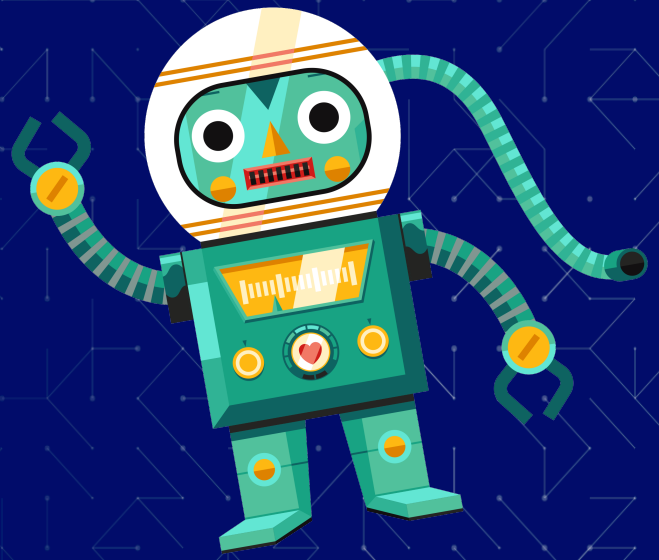
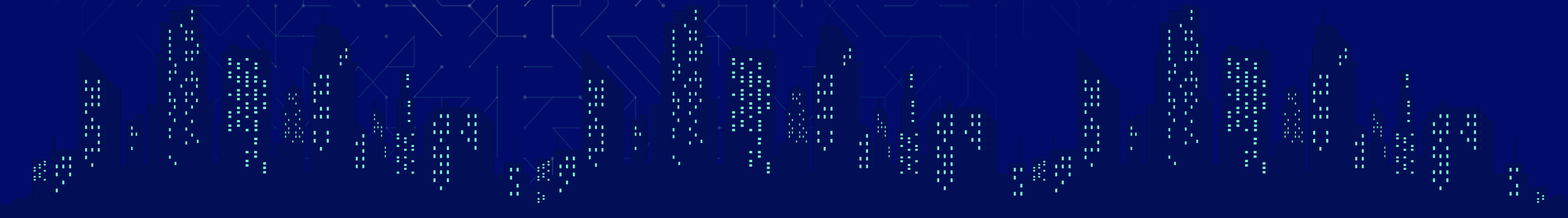


TECHORAMA

TECHORAMA

Modernizing .NET Applications with Docker Containers

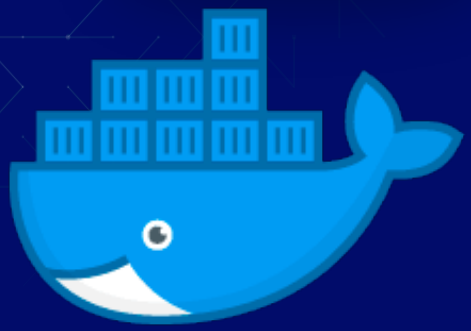
Steven Follis
Docker Inc.
steven.follis@docker.com



Howdy!

I’m Steven Follis:

- Solutions Engineer @ Docker Inc.
- Working with customers with both greenfield & brownfield applications
- Will be smuggling as much chocolate back to The States as possible post-Techorama BE

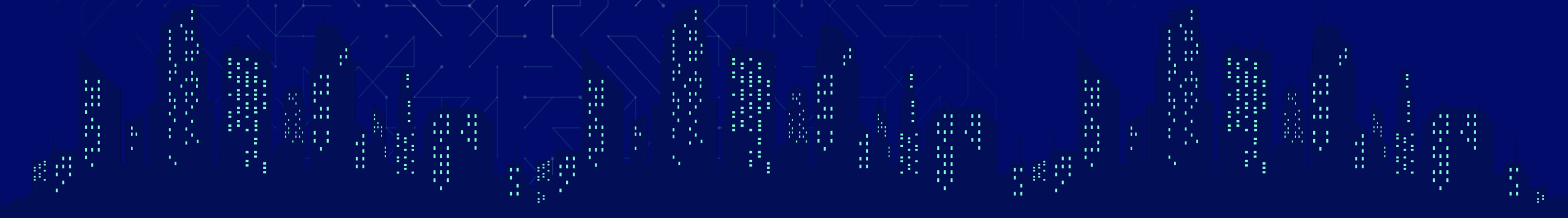


docker

Today

Docker 101	Why Modernize?	Modernization

Docker 101

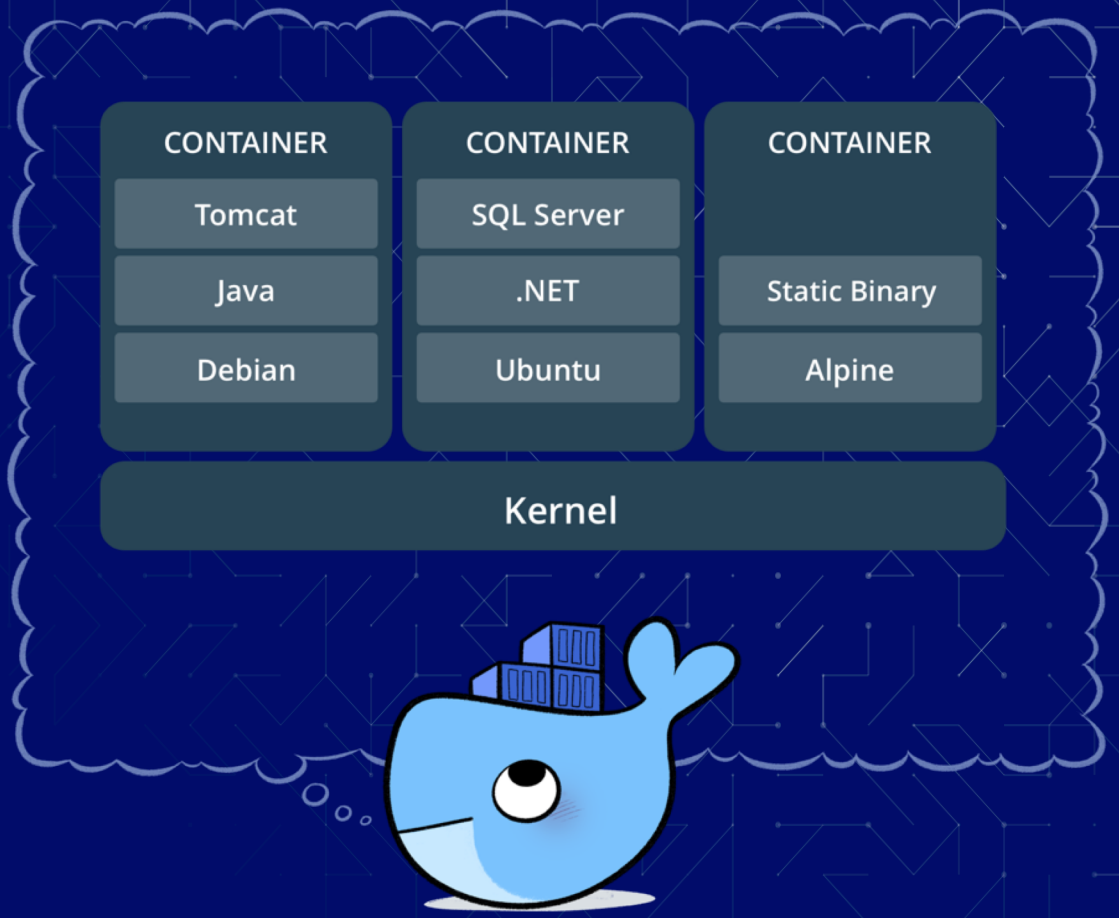


The Container Metaphor

From custom and chaotic to standardized and predictable

DevOps circa 1912

What is a container?



Standardized packaging for software and dependencies

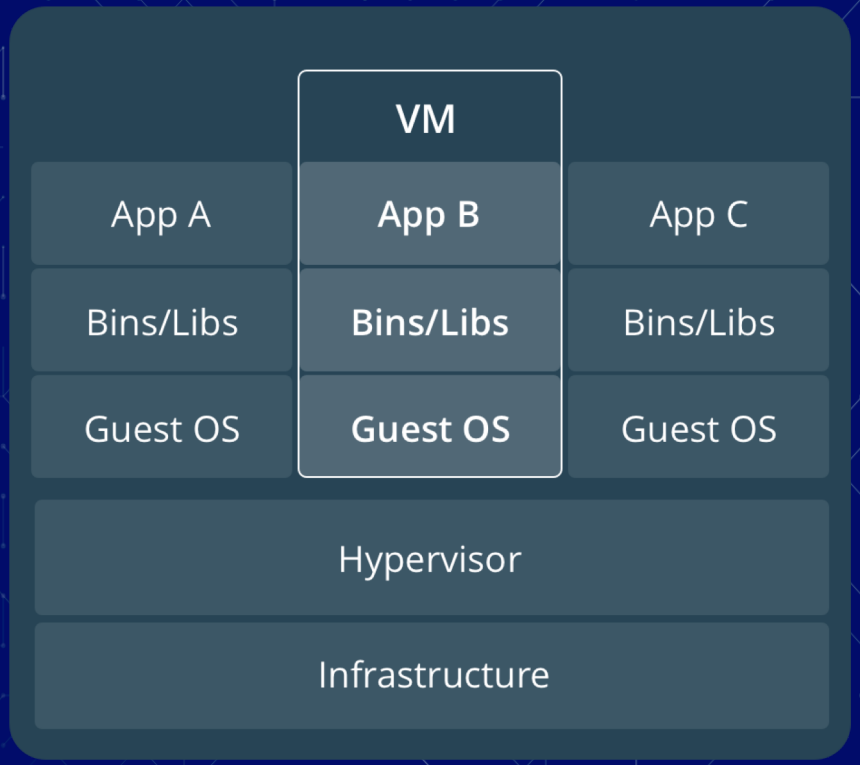
Isolates apps from one another

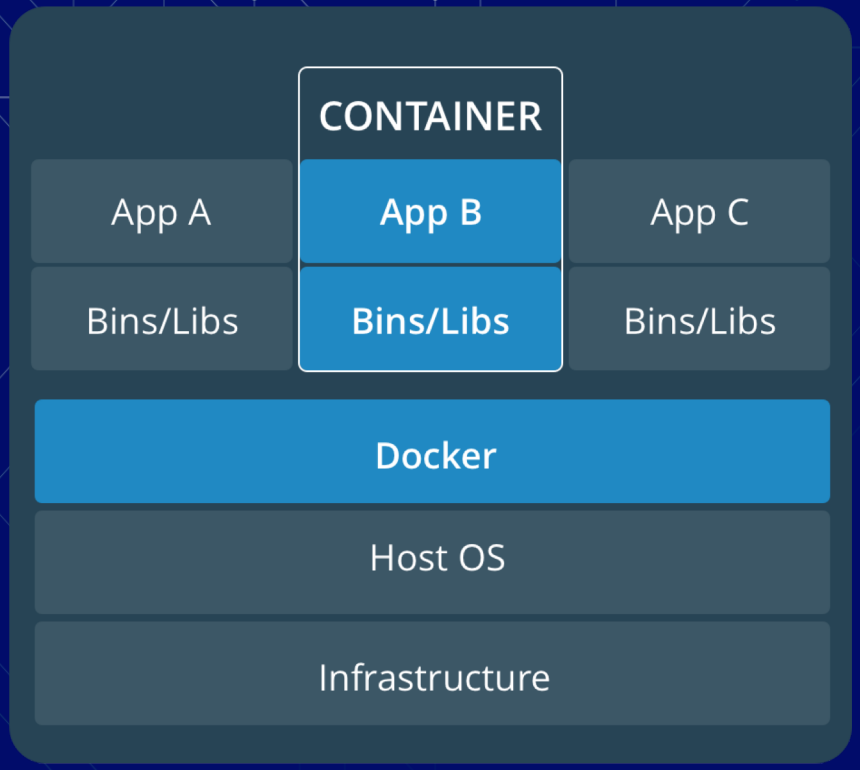
Shares the same OS kernel

Works for all major Linux distributions

Native to Windows Server 2016+

Comparing containers to VMs

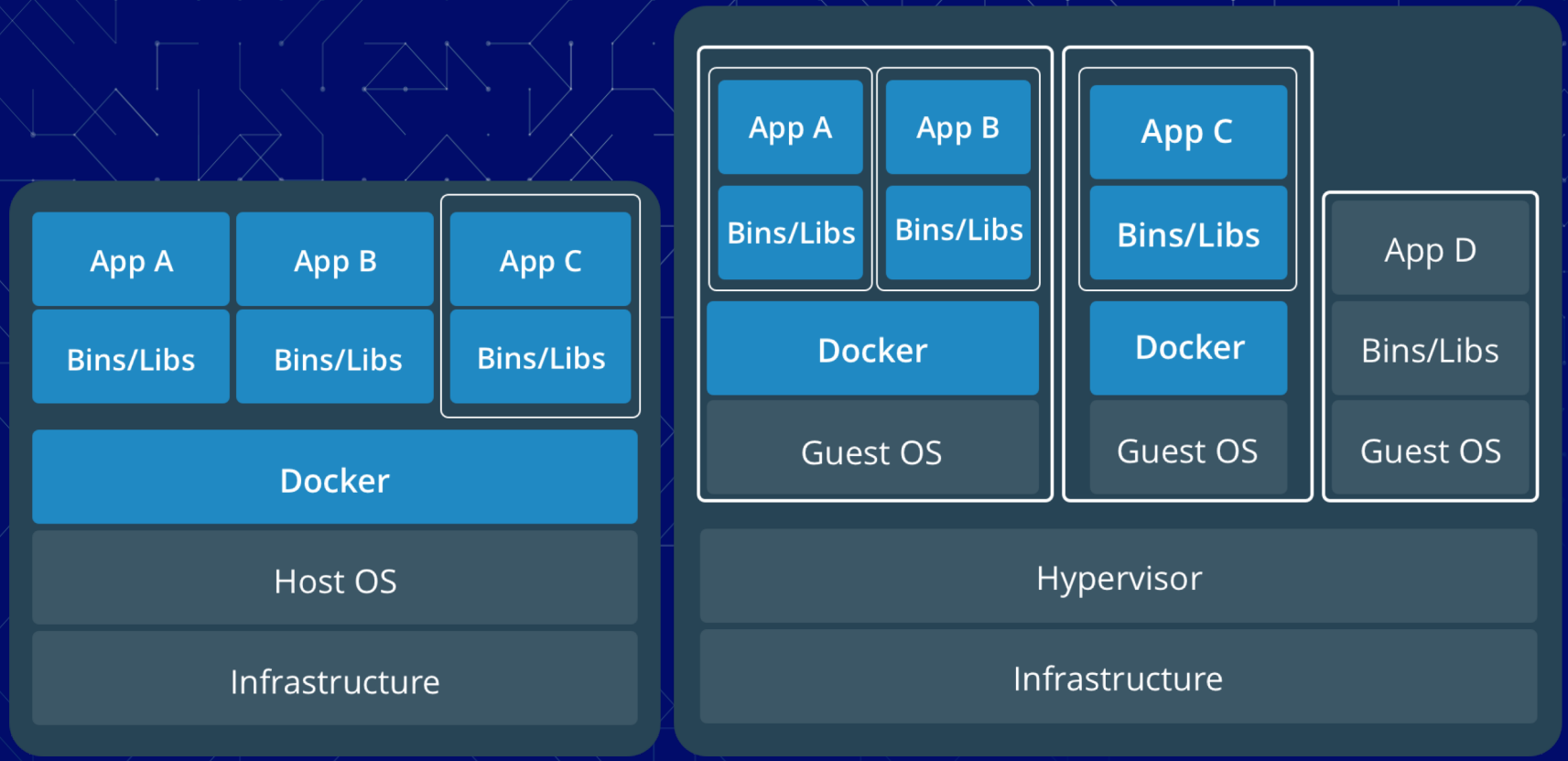




VMs are an infrastructure level construct to turn one machine into many servers

Containers are an app level construct

Containers and VMs together

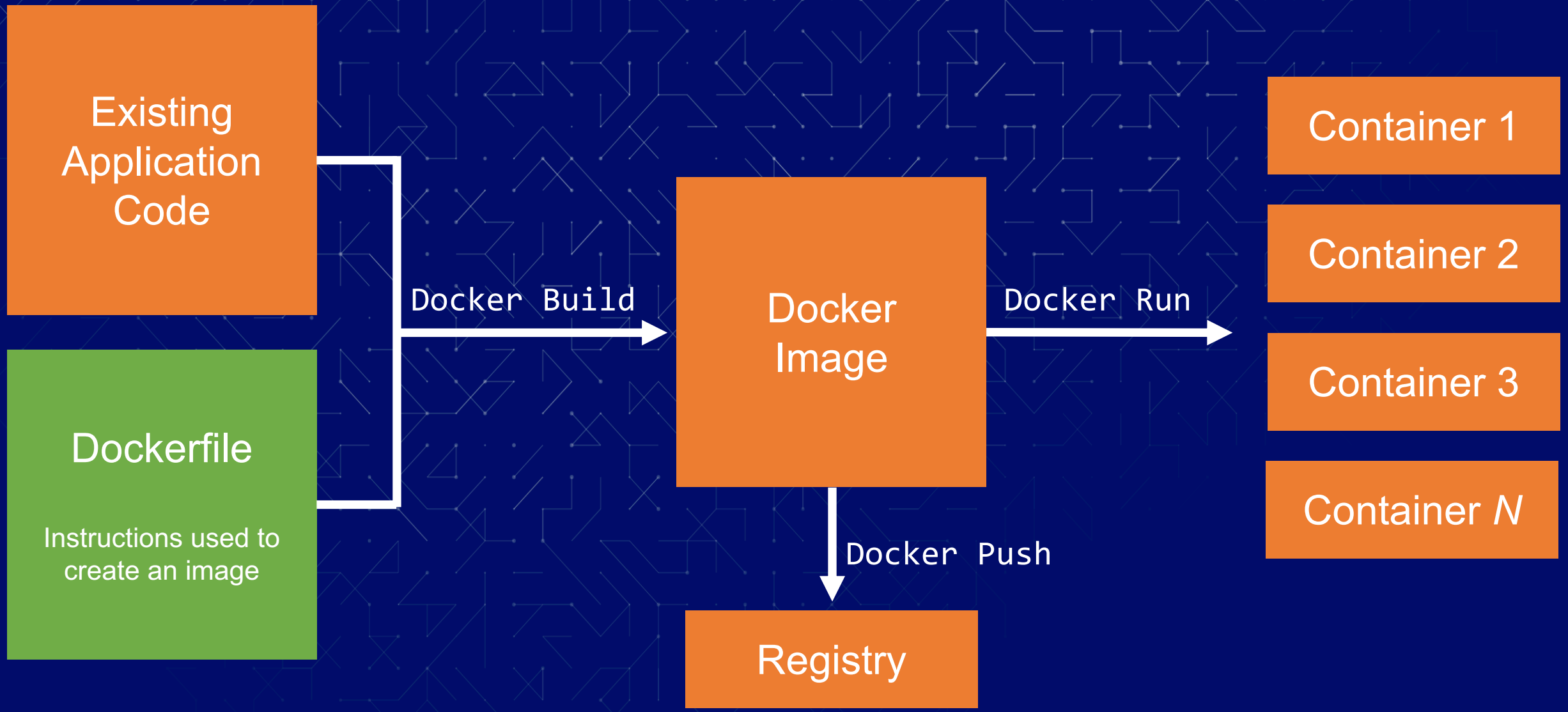


Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

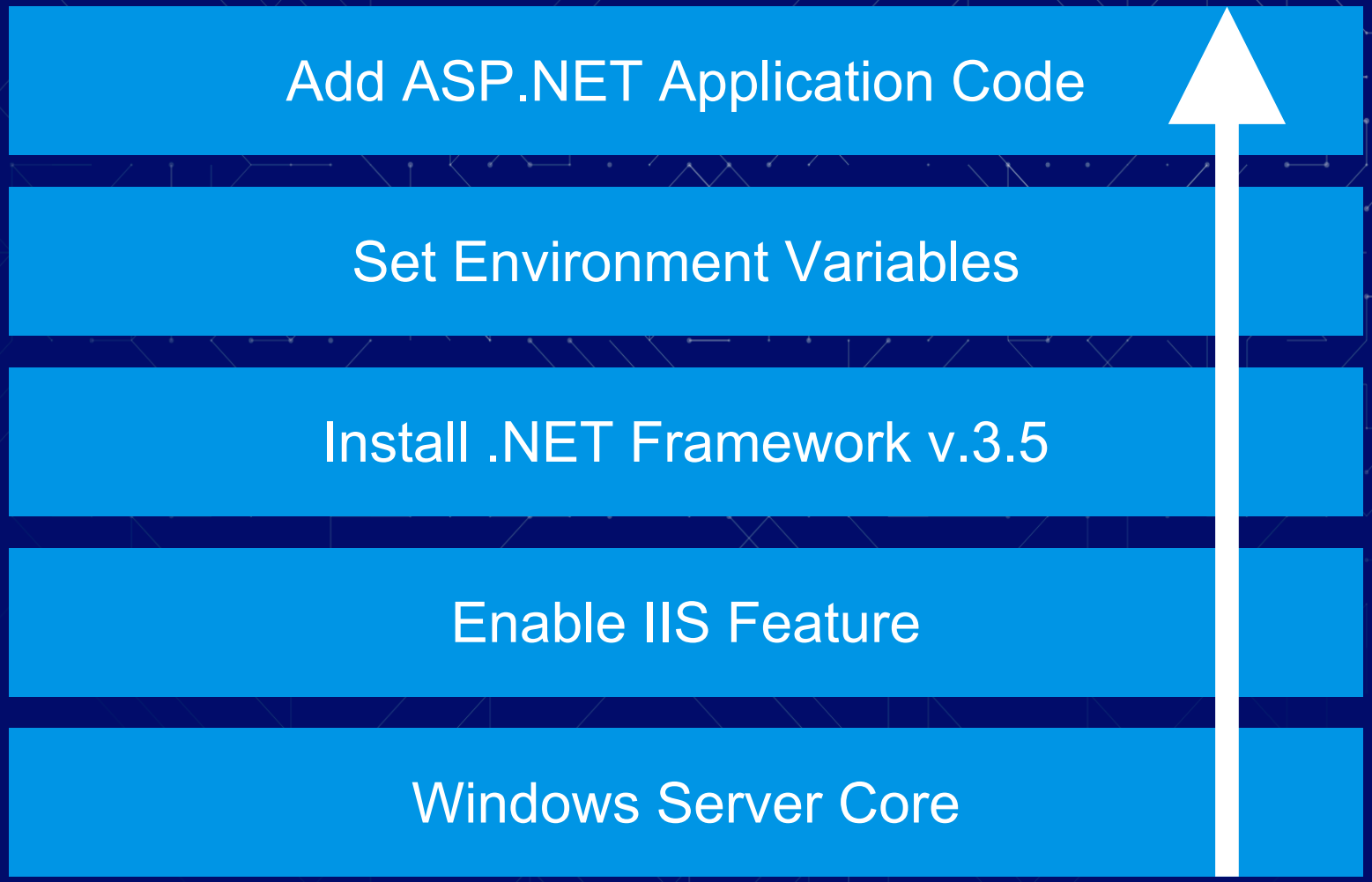
Container isolation options

Windows Server Containers		Hyper-V Containers
Kernel is shared amongst all running containers and host	Kernel	Kernel isolation between each container and the host
No considerations	Licensing	Maximum 2 instances for Windows Server Standard; unlimited for Windows Server Datacenter
Highest number of containers per server	Density	4-6x more memory, impacting number of containers per host

Creating Docker Containers



Docker Images are created in layers



Use Dockerfiles for individual containers

```
FROM    microsoft/aspnet:windowsservercore-10.0.14393.1198

RUN     Set-ItemProperty –path 'HKLM:\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters' -Nam

RUN     Remove-Website -Name 'Default Web Site';`
        New-Item -Path 'C:\web-app' -Type Directory; `
        New-Website -Name 'web-app' -Port 80 -PhysicalPath 'C:\web-app’

HEALTHCHECK --interval=5s ` CMD powershell -command ` try { `$response = iwr http://localhost/Sig

ENV     MESSAGE_QUEUE_URL="nats://message-queue:4222" `
        DB_MAX_RETRY_COUNT="5" `
        DB_MAX_DELAY_SECONDS="10”

ENTRYPOINT ["powershell", "C:\\bootstrap.ps1"]

COPY    .\docker\web\bootstrap.ps1 C:\
COPY    --from=builder C:\out\web\SignUpWeb\_PublishedWebsites\SignUp.Web C:\web-app
```

Use Docker Compose for multiple containers

```
version: '3.3'
services:
        signup-db:
                image: microsoft/mssql-server-windows-express
                environment:
                        - ACCEPT_EULA="Y"
                env_file:
                        - db-credentials.env
                networks:
                        - app-net
        signup-app:
                image: dockeronwindows/ch10-newsletter-web
                env_file:
                        - db-credentials.env
                depends_on:
                        - signup-db
                        - message-queue
                networks:
                        - app-net
        message-queue:
                image: nats:nanoserver
                networks:
                        - app-net
```

The Docker Family Tree

Open source **framework** for assembling core components that make a container platform

Intended for:
Open source contributors + ecosystem partners

Enterprise Edition

Subscription-based, commercially supported **products** for delivering a secure software supply chain

Intended for:
Production deployments + Enterprise customers

Community Edition

Free, community-supported **product** for delivering a container solution

Intended for:
Developers and small teams
Software dev & test

Container orchestration

Scheduling

Affinity/anti-affinity

Health monitoring

Failover

Scaling

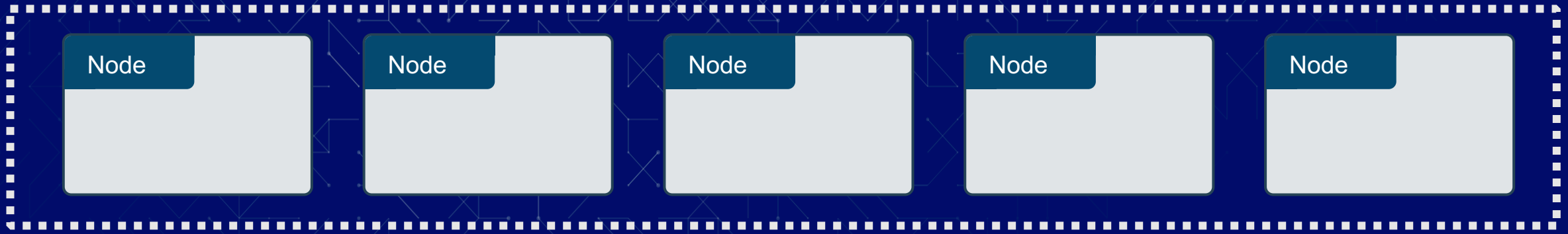
Networking

Service discovery

Coordinated app upgrades

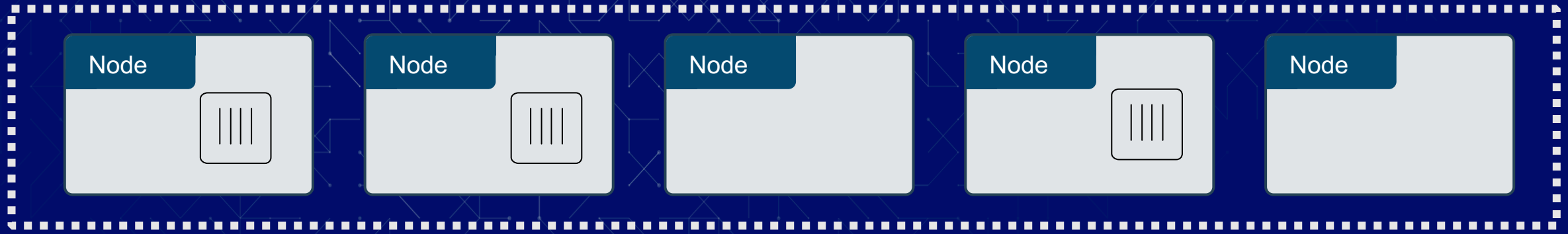
Container orchestration

I want to create 3 copies of a container on my cluster



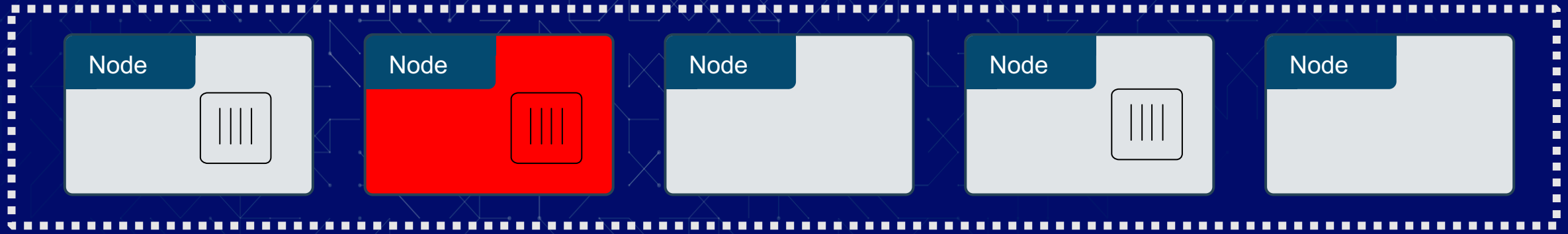
Container orchestration

Three containers are created and spread across the cluster



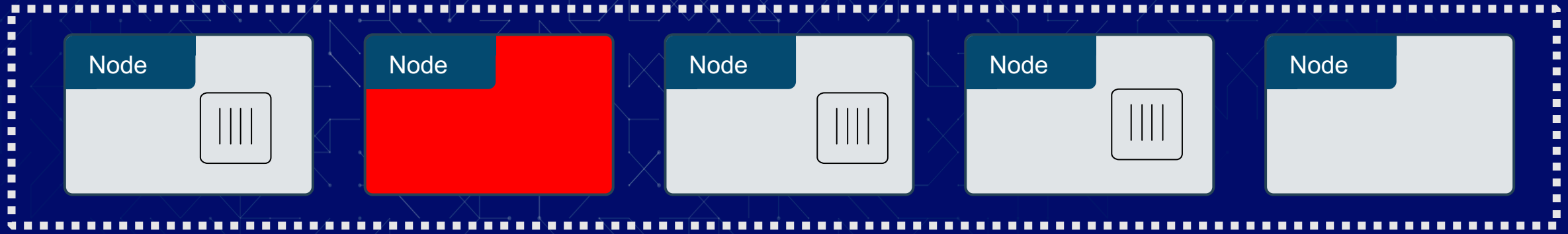
Container orchestration

One node crashes, impacting the container



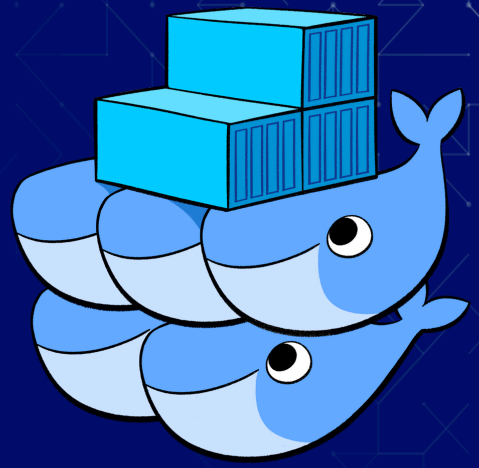
Container orchestration

Orchestrator re-creates a container on a healthy node to maintain 3 copies

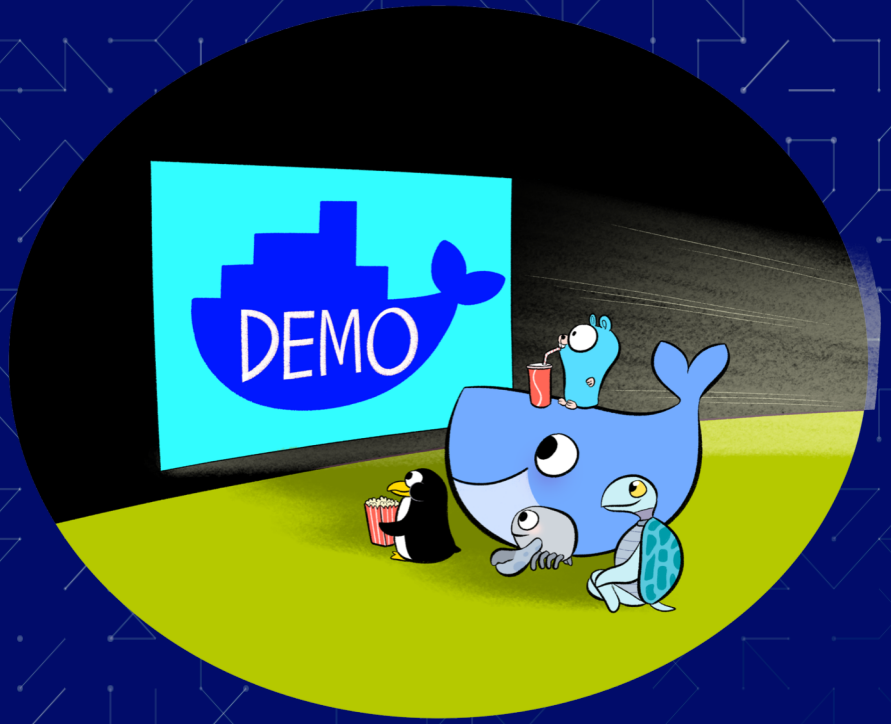


Container orchestration options

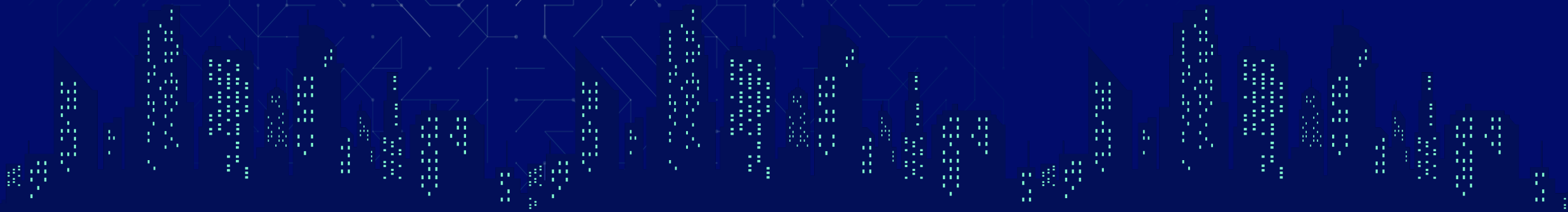
Kubernetes
Advanced capabilities with a broad community



Swarm
Quickly get started today with all workloads, especially Windows


DEMO

Why Modernize?



Maintaining legacy is a drain on innovation

80% IT Budget

spent on maintenance & upkeep

Stuck Keeping the Lights on

Making it difficult to keep up with accelerating standards

Application Changes

become too complex, difficult, and/or costly to implement

Windows Server 2008 is nearing end of life

January 14, 2020
is just months away

End of standard support

End of security patches

End of hotfixes

Options for migrating off of Windows Server 2008

Refactor and upgrade

- Requires engineering resources
- Depending on familiarity with the app and complexity of the app, can take several weeks per application
- Once upgraded, need to repeat the same process in a few years

Extended support contract

- Expensive: can cost as much as 75% of license cost per year
- Kicking can down the road; will need to upgrade eventually so why wait?

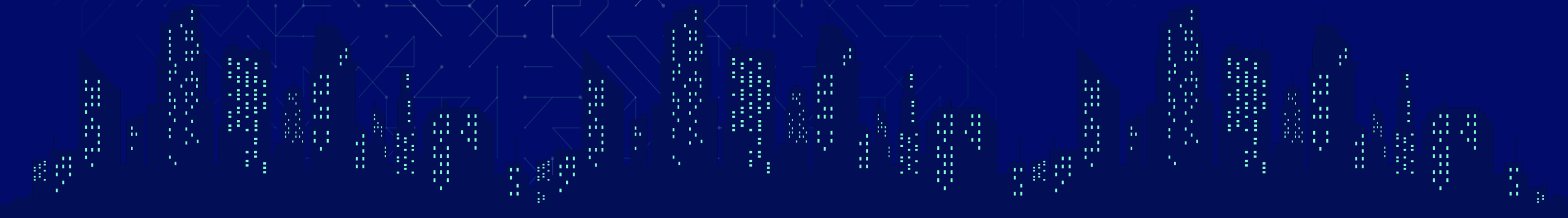
“Lift and shift” servers to the public cloud

- If running on vSphere on-prem, will require a full app conversion which can take several weeks
- Only delays the inevitable as app will still be running an older OS

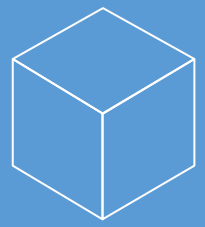
Containerize with Docker

- Upgrade to the latest Windows Server versions 2016, 1709, 1803, WS2019
- Gain cloud portability and choice of where to deploy the app
- Future-proof apps to simplify upgrades forever

Modernizing Applications



Classifying applications

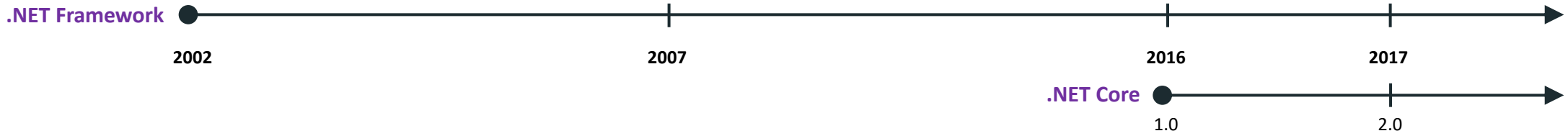


Legacy

Brownfield

Greenfield

.NET Implementations



- Rock solid framework and 3rd party ecosystem
- Windows Only
- Containers based on Windows Server Core

- Modular framework inspired by NodeJS, Ruby, etc.
- Cross platform (Windows, OSX, Linux)
- Containers based on Windows Server Nano

Choosing a base layer

Nano Server	Windows Server Core	Windows
`mcr.microsoft.com/windows/nanoserver:1809`	`mcr.microsoft.com/windows/servercore:1809`	`mcr.microsoft.com/windows:1809`
Greenfield and cloud-native applications	Brownfield and Legacy applications	Carries most Windows OS components
.NET Core	Full .NET Framework	Automation workloads
94 MB	1.4 GB	3.5 GB

*based on Windows Server 2019

History of .NET

Version	Release	Development tool	Included in	
			Windows	Windows Server
1.0	2002-02-13	Visual Studio .NET	XP	N/A
2.0	2005-11-07	Visual Studio 2005	N/A	2003, 2003 R2, 2008 SP2
3.0	2006-11-06	Expression Blend	Vista	2008 SP2, 2008 R2 SP1
3.5	2007-11-19	Visual Studio 2008	7, 8, 8.1, 10	2008 R2 SP1
4.0	2010-04-12	Visual Studio 2010	N/A	N/A
4.5	2012-08-15	Visual Studio 2012	8	2012
4.6	2015-07-20	Visual Studio 2015	10	N/A
4.7	2017-04-05	Visual Studio 2017	10 v1703	N/A

Beginning with Windows Containers

- Architecture
- Runtime
- Dependencies
- Implementation

- Select 1-3 target applications
- Containerize-able in 2-3 days of work
- Representative of application portfolio
- Aligned with existing initiatives
- Technical resources/app owner available

Beginning with Windows Containers

- Architecture
- Runtime
- Dependencies
- Implementation

- .NET Framework or Java EE
- 2-3 tier architecture
- 1-5 runtime components
- Manageable / known dependencies

Beginning with Windows Containers

Architecture

Runtime

Dependencies

Implementation

- IIS 6-8
- .NET Framework 2.0 or later
- Tomcat, WebLogic, WebSphere, JBoss

Beginning with Windows Containers

- Architecture
- Runtime
- Dependencies
- Implementation

- Availability of application artifacts + expertise
- Databases out of scope for initial containerization
- Consider database and user authentication

Beginning with Windows Containers

- Architecture
- Runtime
- Dependencies
- Implementation

- No hardcoded IPs or hostnames
- Application startup time < ~5 minutes
- Deployment artifacts support unattended installation

Commercial off the shelf software (COTS)

Containerizing COTS applications vary widely by vendor depending on:

- Technology components
- Licensing considerations
- Supportability

Must support unattended installation & configuration – .msi, .exe

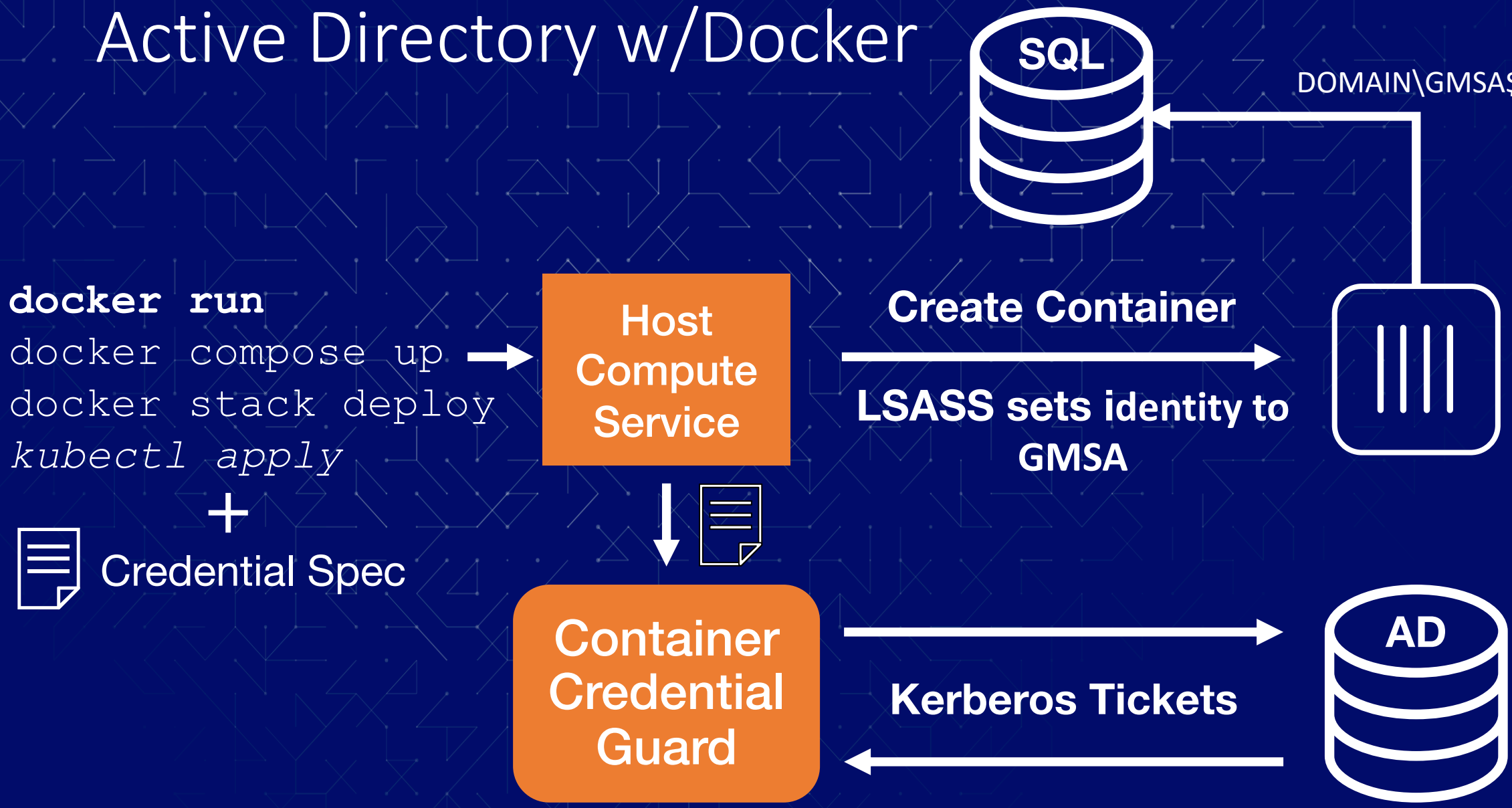
Many vendors providing Docker Certified images of their software

Microsoft ORACLE IBM SAP splunk>

SQL in Containers

- Microsoft offers SQL Server in Linux and Windows containers
- “Real” SQL, including support for traditional tools such as SQL Server Management Studio, dacpac & bacpac deployments
- Scenarios
  - Dev/Test environments
  - Quality assurance teams
  - Automated testing workflows
- Consider available RAM
- Best practices still emerging for production workloads

Active Directory w/Docker



Modernization anti-patterns

Desktop client workloads

- Universal Windows Apps
- Windows Forms
- Windows Presentation Framework
- Visual Basic

Low-level Windows APIs unavailable in Windows Server Core

- Ex. WASAPI, Fax Services

Hardware devices

- Coming soon in Docker 19.03

Windows Server Role Matrix

Server Role	Available in Full Installation	Available in Server Core
Active Directory Certificate Services (AD CS)	✓	
Active Directory Domain Services (AD DS)	✓	✓
Active Directory Federation Services (AD FS)	✓	
Active Directory Lightweight Directory Services (AD LDS)	✓	✓
Active Directory Rights Management Services (AD RMS)	✓	
Application Server	✓	
DHCP Server	✓	✓
DNS Server	✓	✓
Fax Server	✓	
File Services	✓	✓
Hyper-V	✓	✓
Network Policy and Access Services	✓	
Print Services	✓	✓
Streaming Media Services	✓	✓
Terminal Services	✓	
UDDI Services	✓	
Web Server (IIS)	✓	✓
Windows Deployment Services	✓	

Windows Server Channels

Long-Term Servicing Channel (LTSC) – Currently *Windows Server 2019*

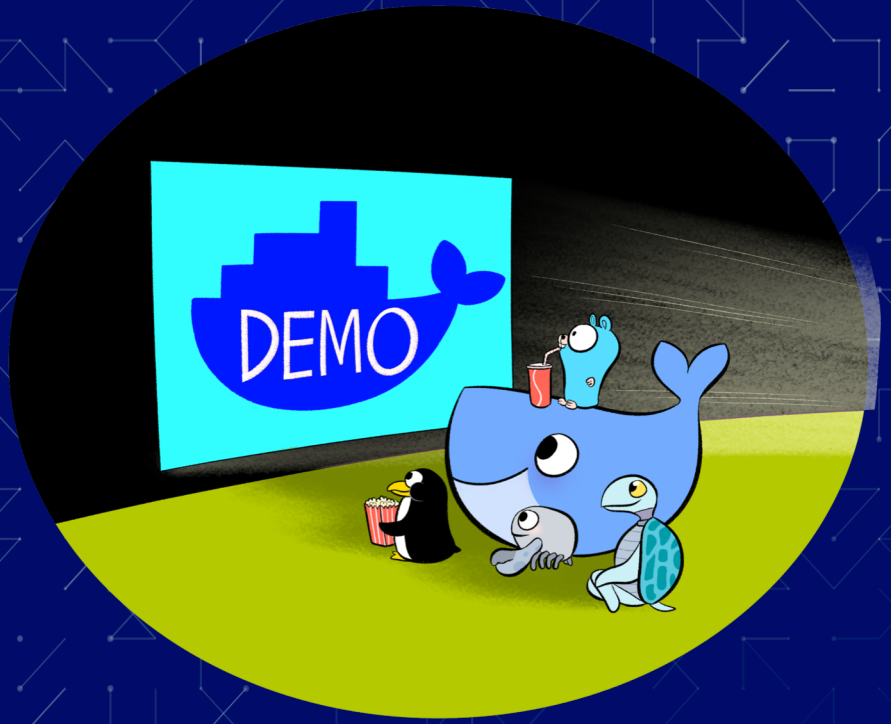
- New major version of Windows Server every 2-3 years
- 5 years of mainstream support + 5 years of extended support
- Stable, predictable

Semi-Annual Channel – Currently *Windows Server, version 1809*

- New versions twice a year (Spring + Fall)
- 18 months of support
- Faster release cadence with latest features
- Most features will be rolled into next LTSC release but not guaranteed
- Requires volume licensing or a cloud provider

Use Windows Server 2019 when possible

- Named Pipe Mounts
- Multiple Containers per gMSA
- Better Hyper-V Isolation
- SMB Global Mapping
- Kubernetes


DEMO

Summary

1

Containerize legacy applications to gain agility & cost savings

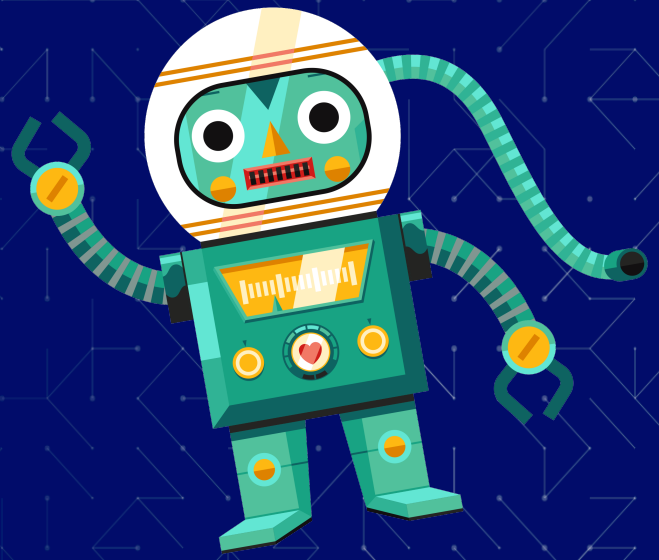
2

Start small & deveop muscle around Docker

3

Consider identity and storage needs early

https://tinyurl.com/techoramadocker



TECHORAMA

Thanks!