



Modernizing .NET Applications with Docker Containers

Steven Follis

steven.follis@docker.com



Today

Docker
101

Why
Modernize?

Modernization

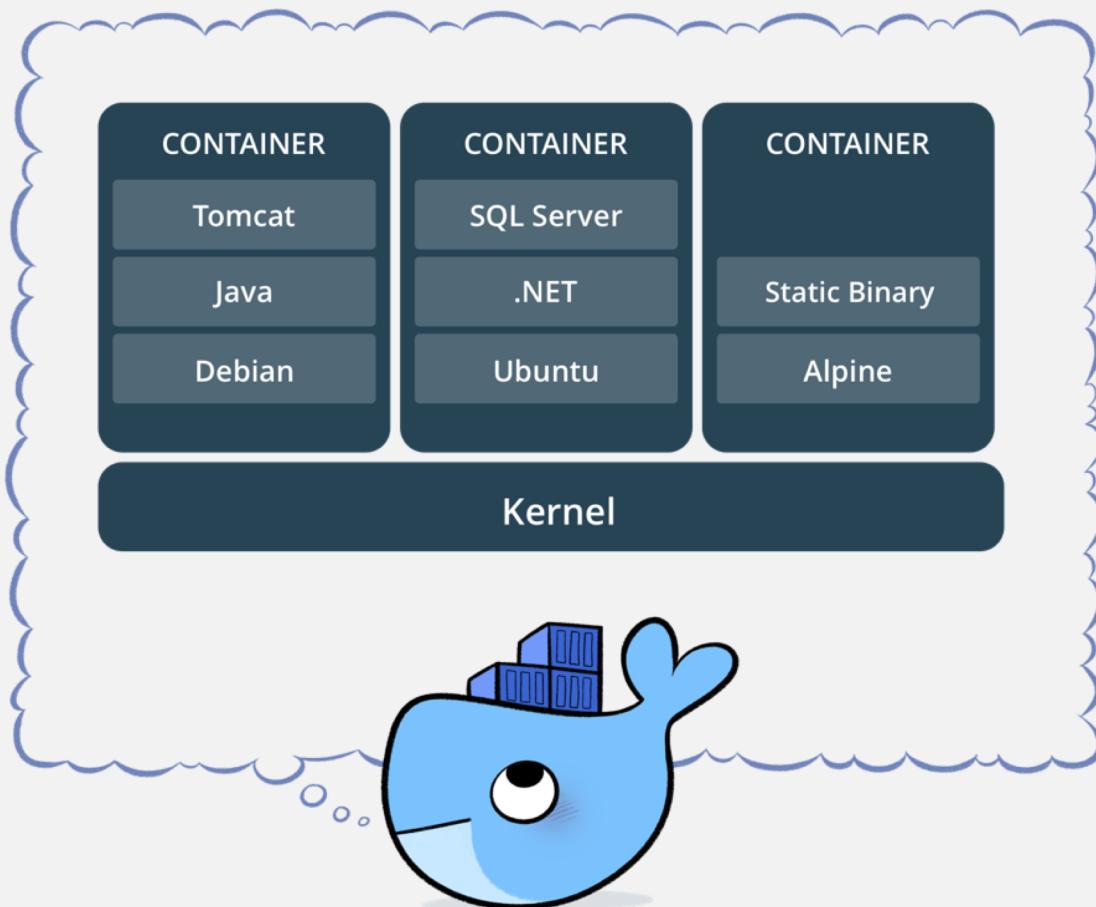
Docker 101

The Container Metaphor

From custom and chaotic to standardized and predictable

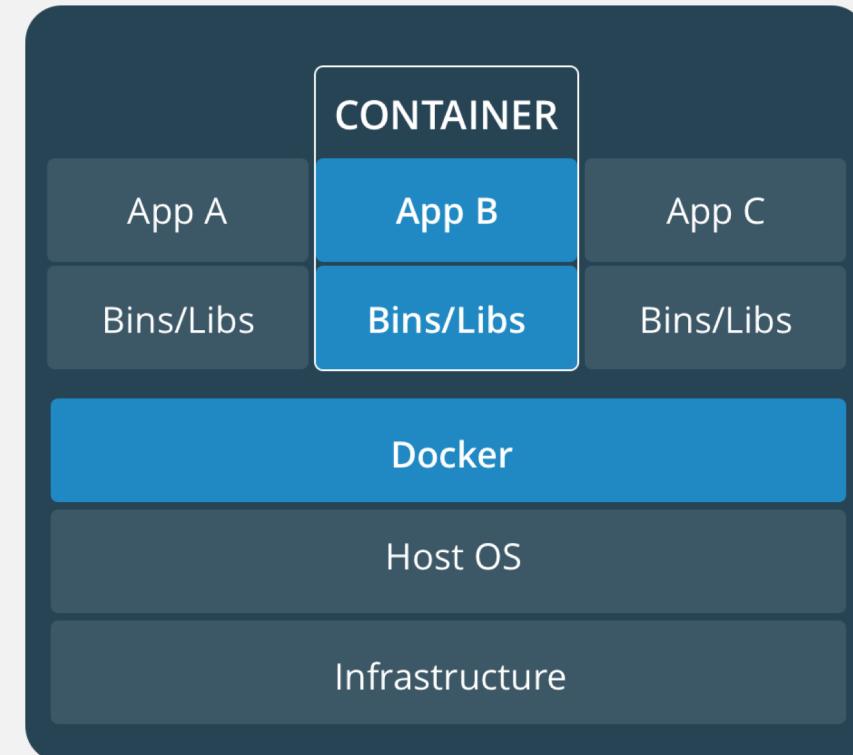
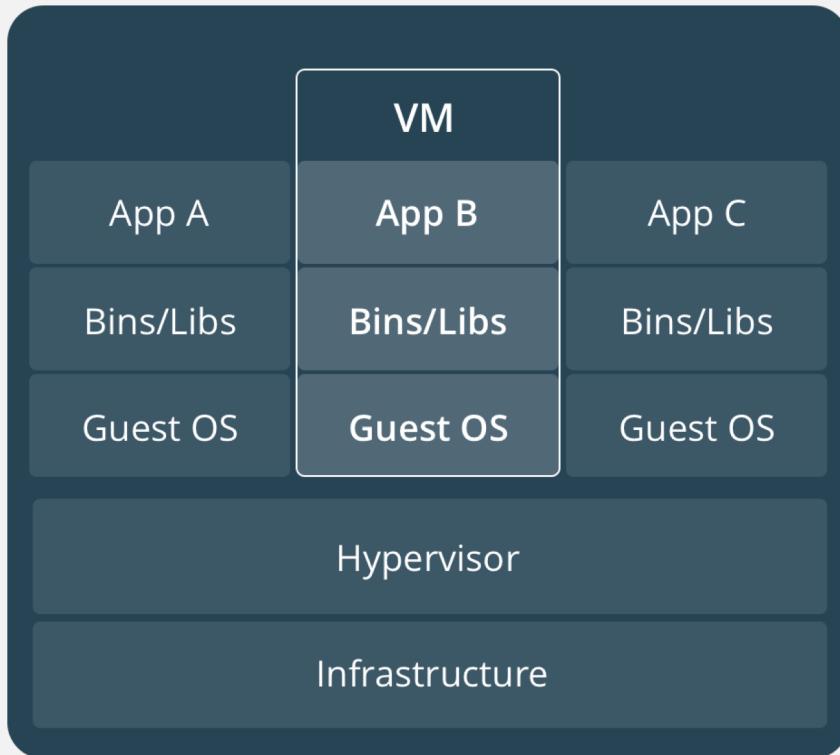


What is a container?



- 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 virtual machines



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

Containers are an app level
construct

Container isolation options

Windows Server Containers

Kernel is shared amongst all running containers and host

Kernel

No considerations

Licensing

Highest number of containers per server

Density

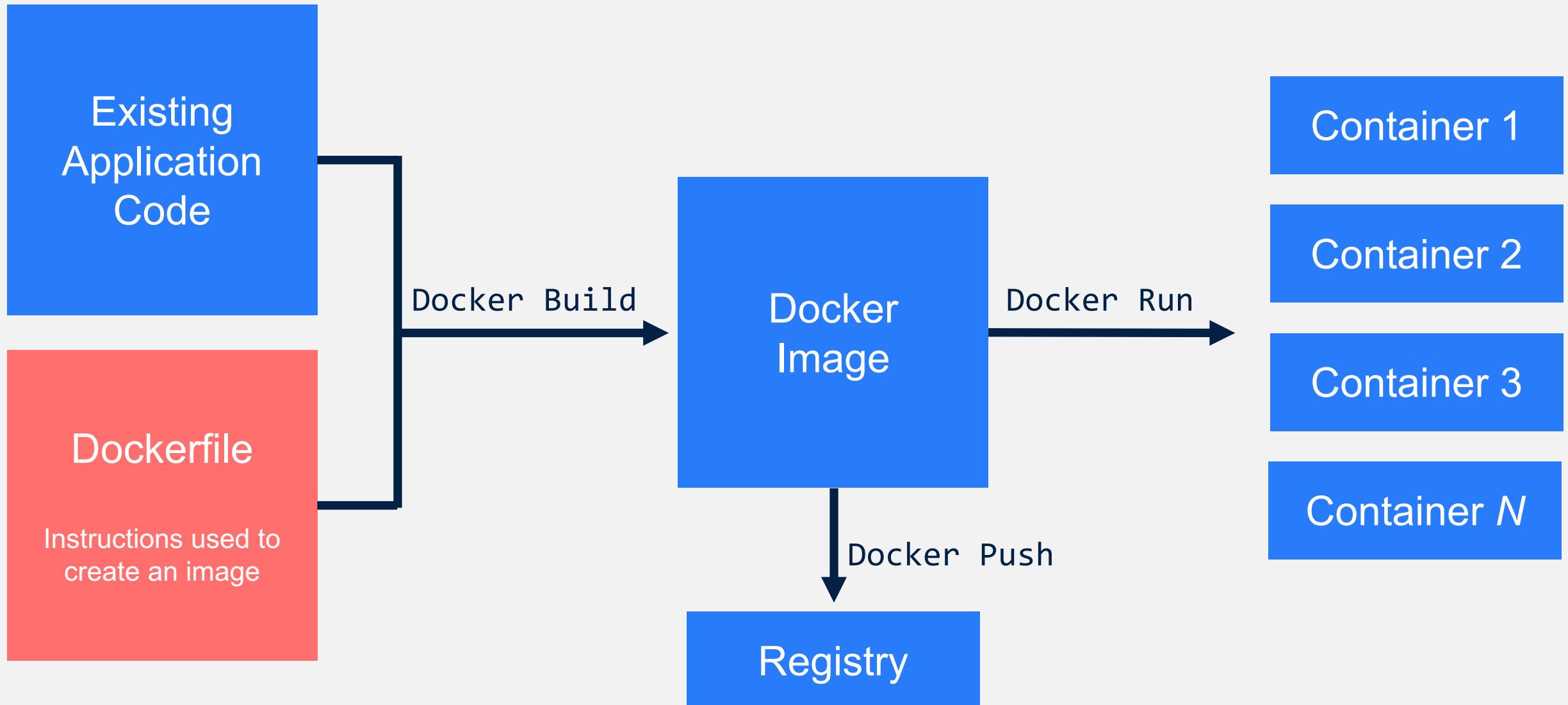
Hyper-V Containers

Kernel isolation between each container and the host

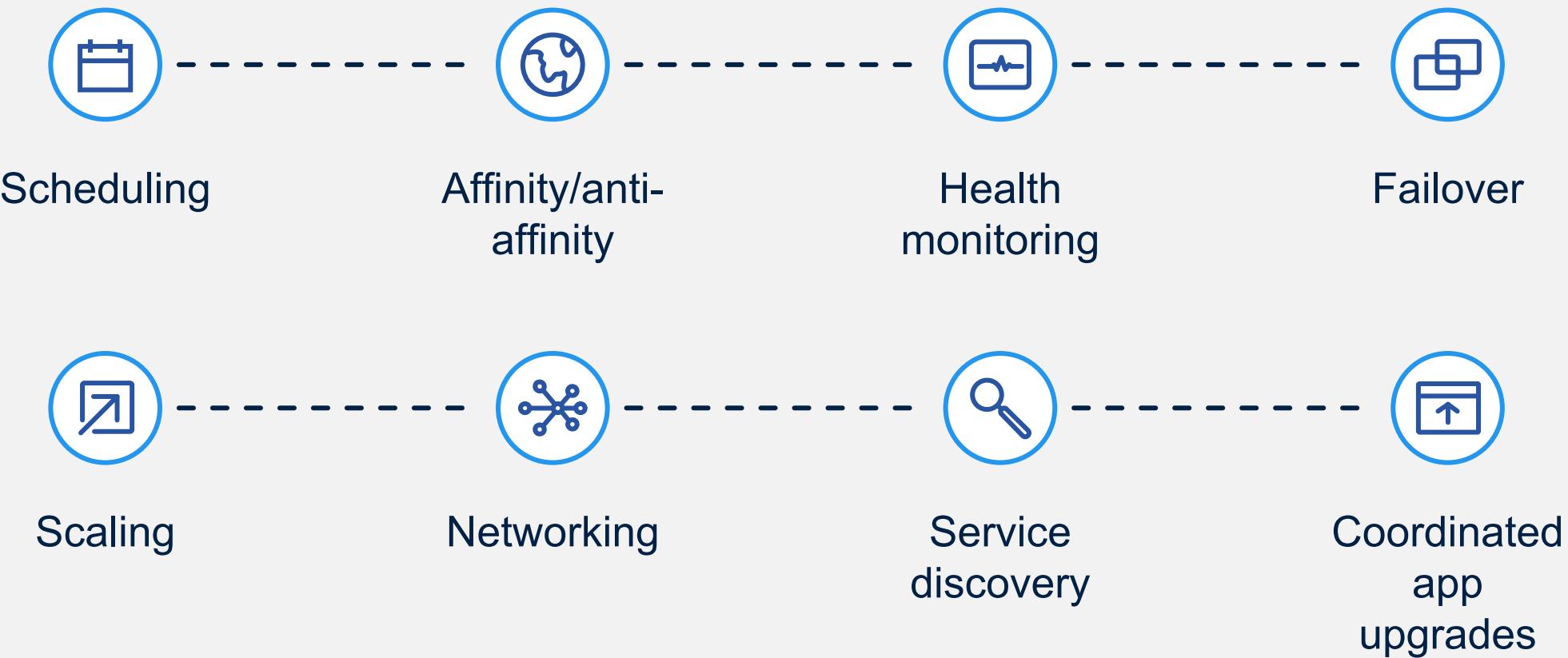
Maximum 2 instances for Windows Server Standard; unlimited for Windows Server Datacenter

4-6x more memory, impacting number of containers per host

Building a container

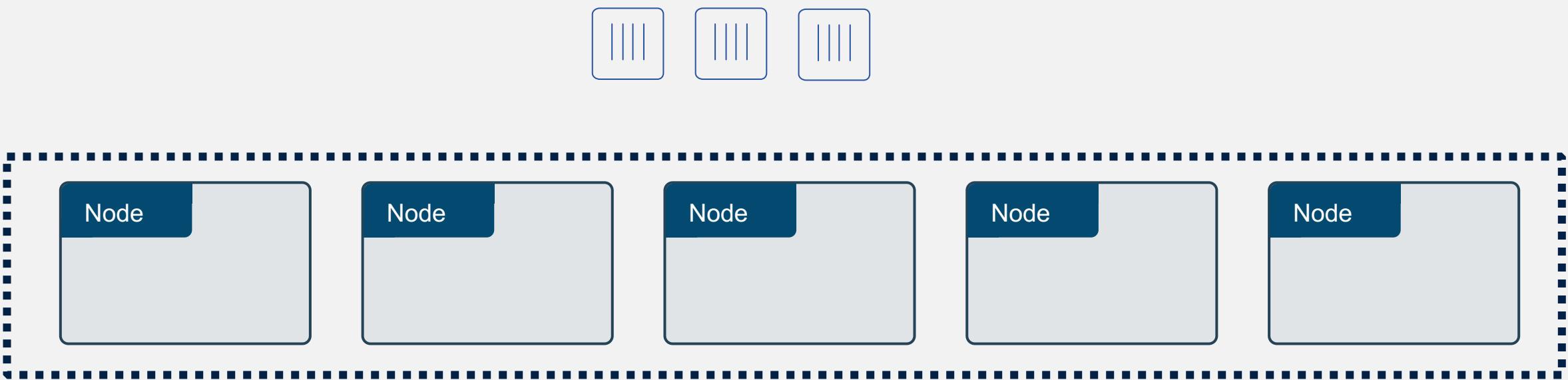


Container Orchestration



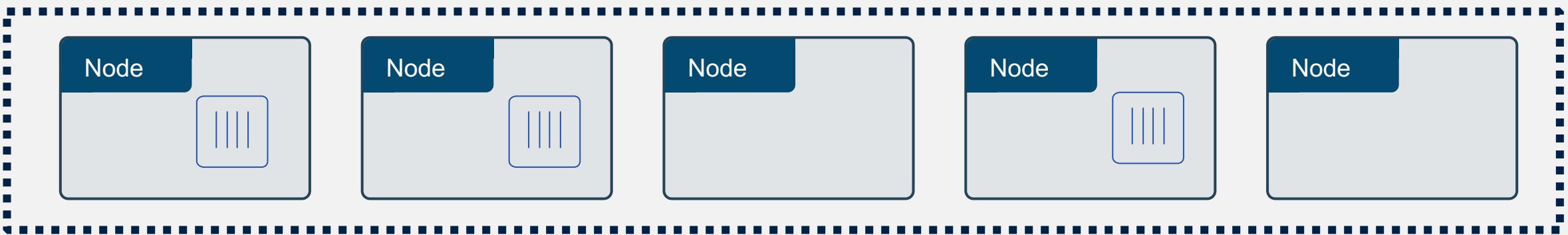
Container Orchestration

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



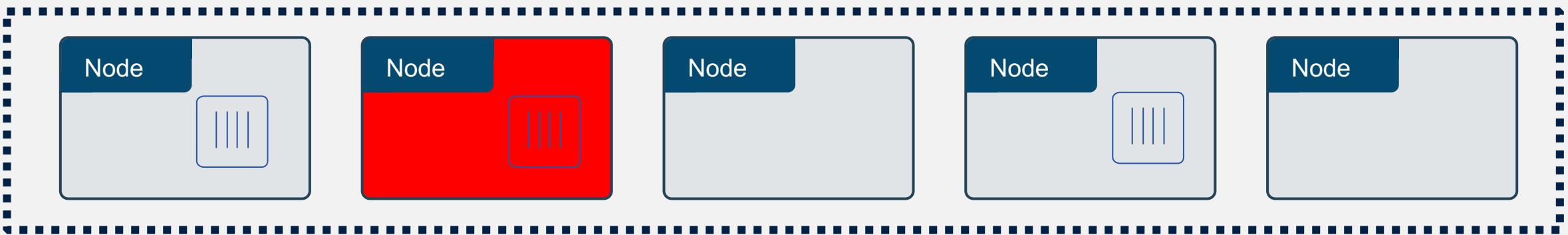
Container Orchestration

Three containers are created and spread across the cluster



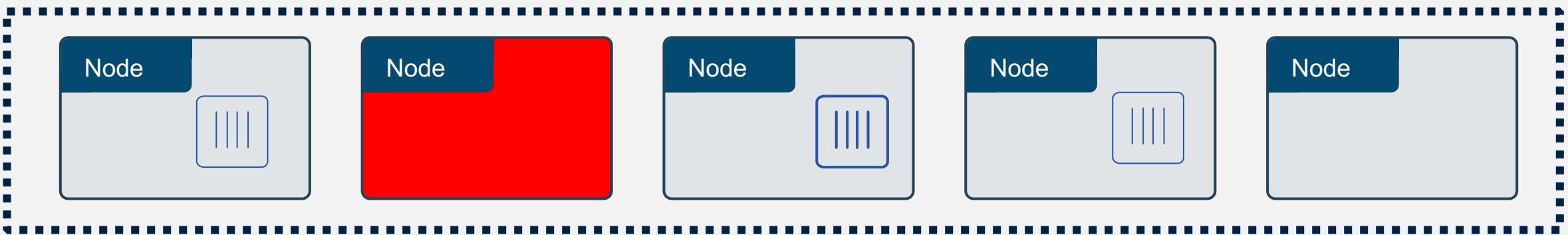
Container Orchestration

One node crashes, impacting a container



Container Orchestration

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

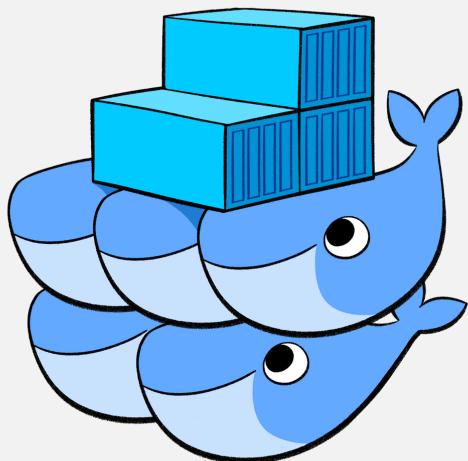


Container Orchestrators



Kubernetes

Advanced capabilities with a broad community

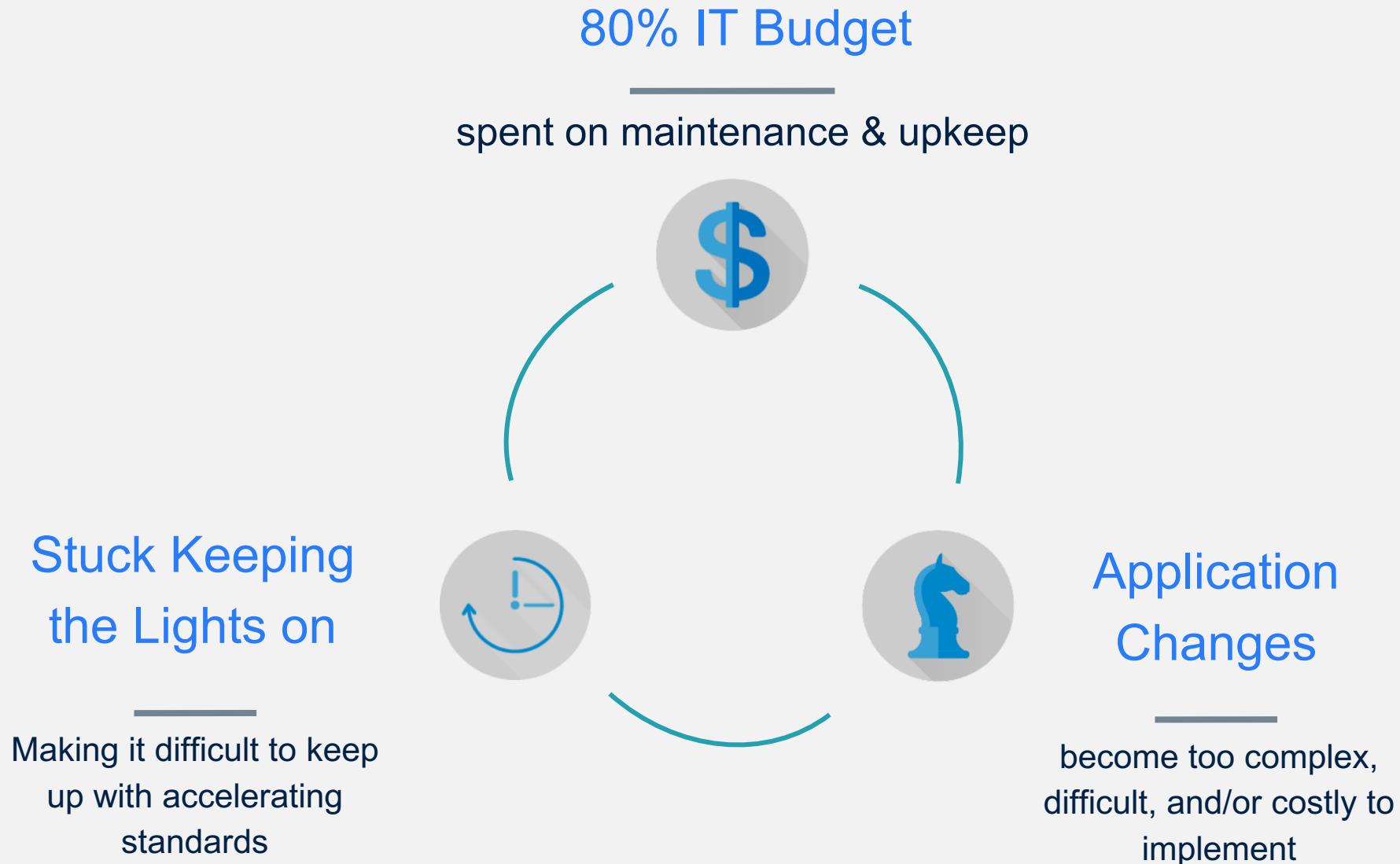


Swarm

Quickly get started today with all workloads,
especially Windows

Why Containerize?

Maintaining legacy is a drain on innovation



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 from 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 & shift” servers
to the public cloud

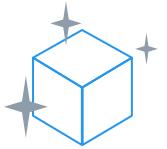
- If running on vSphere on-prem, will require a full app conversion which can take several weeks
- Only delays the inevitable as app will still be running an older OS

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

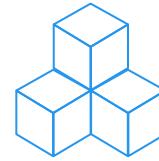
Modernizing Applications

Designed to quickly deliver success at scale



Replatform Legacy

Containerize legacy Java and .NET applications



Modernize Brownfield

Modernize existing applications under active development



Accelerate Greenfield

Develop new container-first applications

Beginning with Windows Containers

Architecture

- 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

Runtime

Dependencies

Implementation

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

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

Runtime

Dependencies

Implementation

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

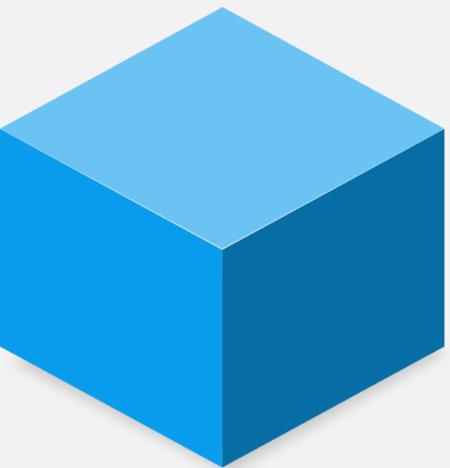
Architecture

Runtime

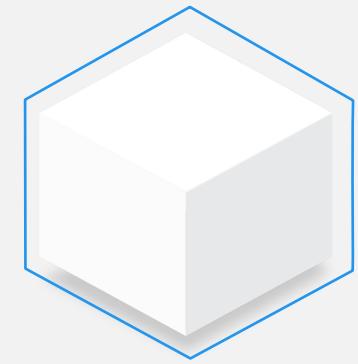
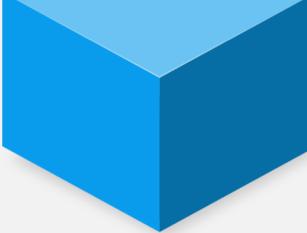
Dependencies

Implementation

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



Demo



Commercial off the shelf software (COTS)

Containerizing COTS applications varies 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



ORACLE®



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

Persistent Storage

1

Use SMB Global Mapping to Mount network share to host machine

2

Mount path during container initialization

```
$creds = Get-Credential  
New-SmbGlobalMapping `br/>    -RemotePath \\contosofileserver\share1 `br/>    -Credential $creds `br/>    -LocalPath G:
```

Active Directory with Docker

docker run

docker compose up →

docker stack deploy

kubectl apply

+



Credential Spec



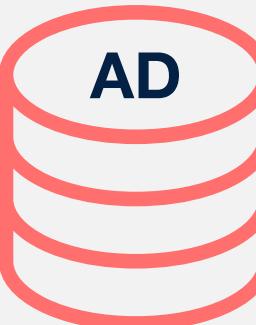
Create Container
→
LSASS sets identity to GMSA



DOMAIN\GMSA\$



→
Kerberos Tickets
←



Windows Server Servicing 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 1903*

- 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

Today



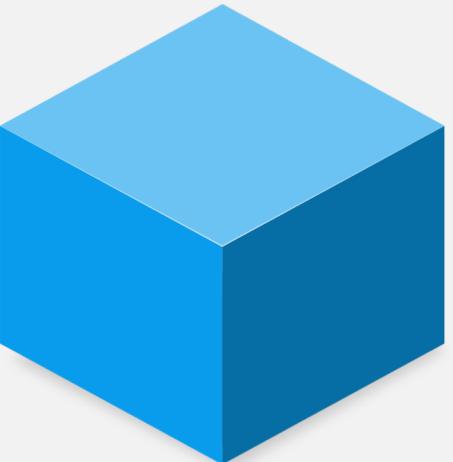
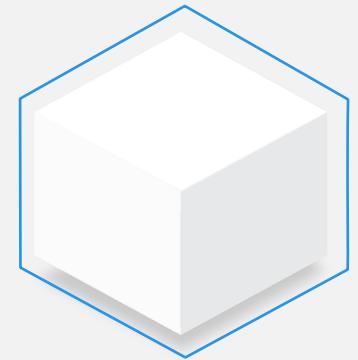
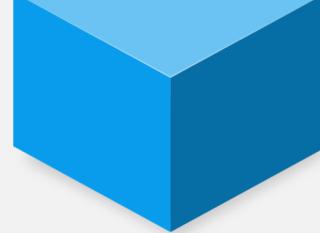
Containerize legacy
applications to gain
agility & cost savings



Start small &
develop muscle
around Docker



Consider identity
and storage needs
early



Thank you



Come see us at ATO Booth #14