https://github.com/marrobi/Microsoft-and-Containers

Microsoft

Enabling DevOps practices with Containers, Docker, Linux, Windows and Microsoft Azure

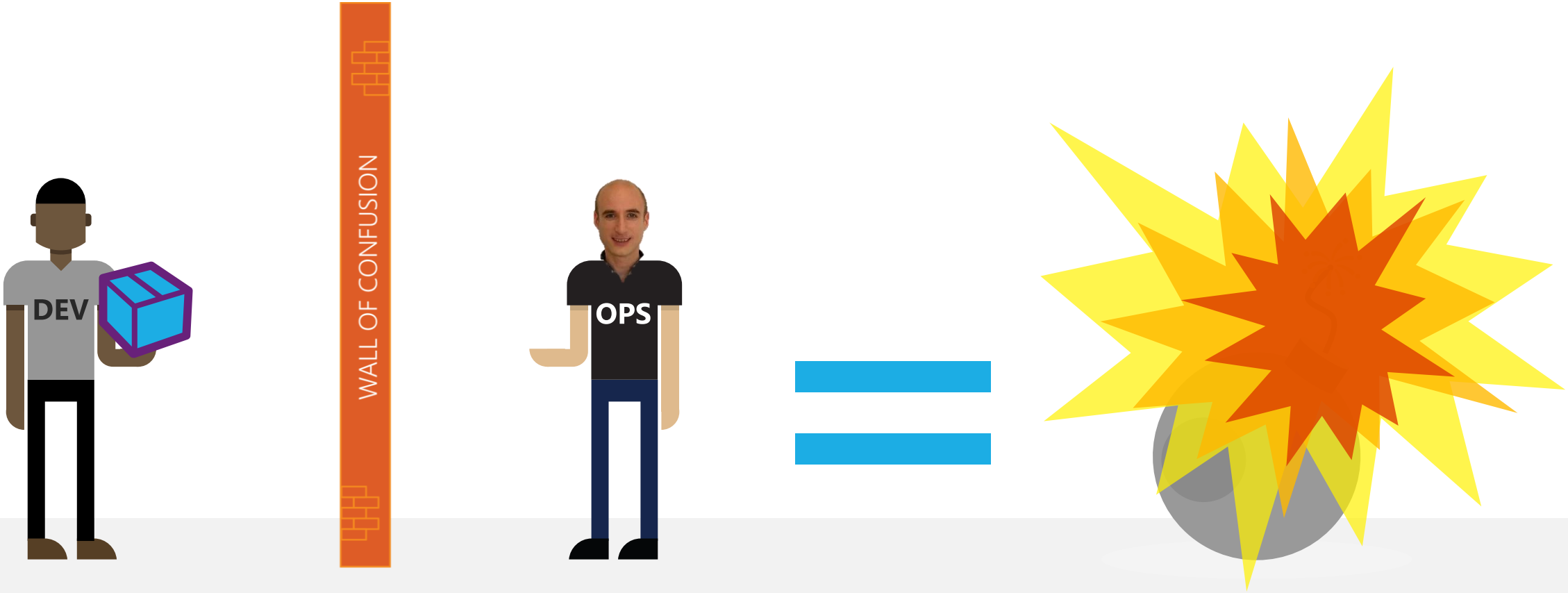
Marcus Robinson
Technical Evangelist
@techdiction

What are we going to talk about?

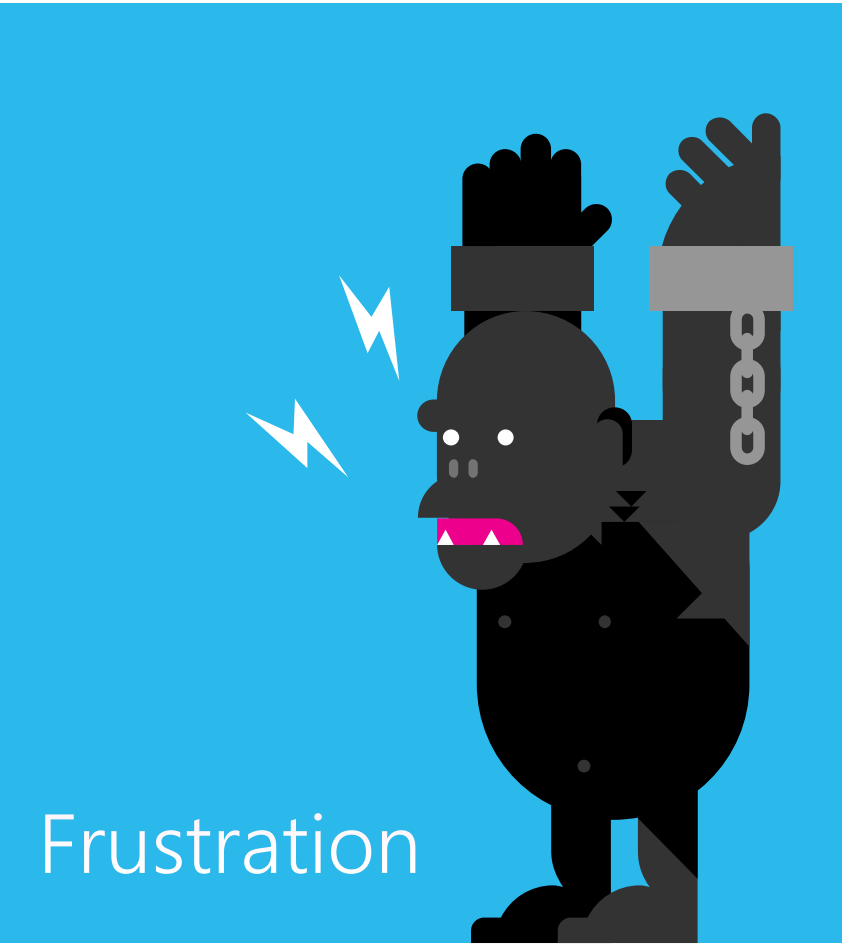
- DevOps
- Containers
- Docker
- Windows Containers
- Containers on Azure
- Visual Studio Team Services

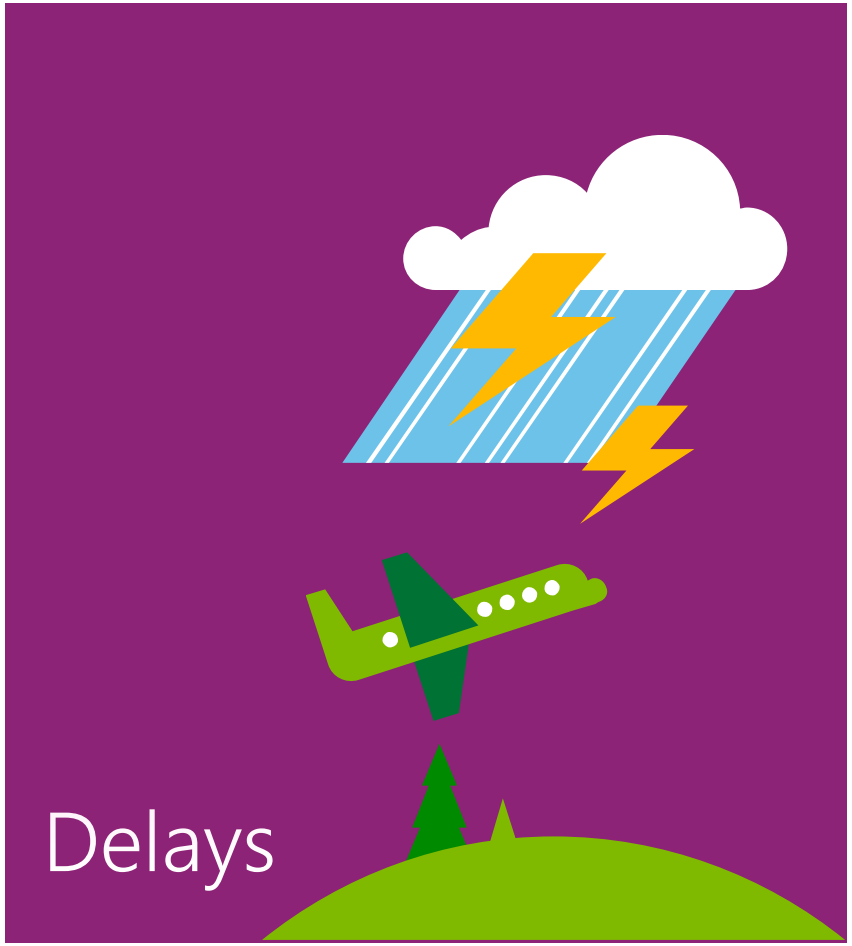
What is DevOps

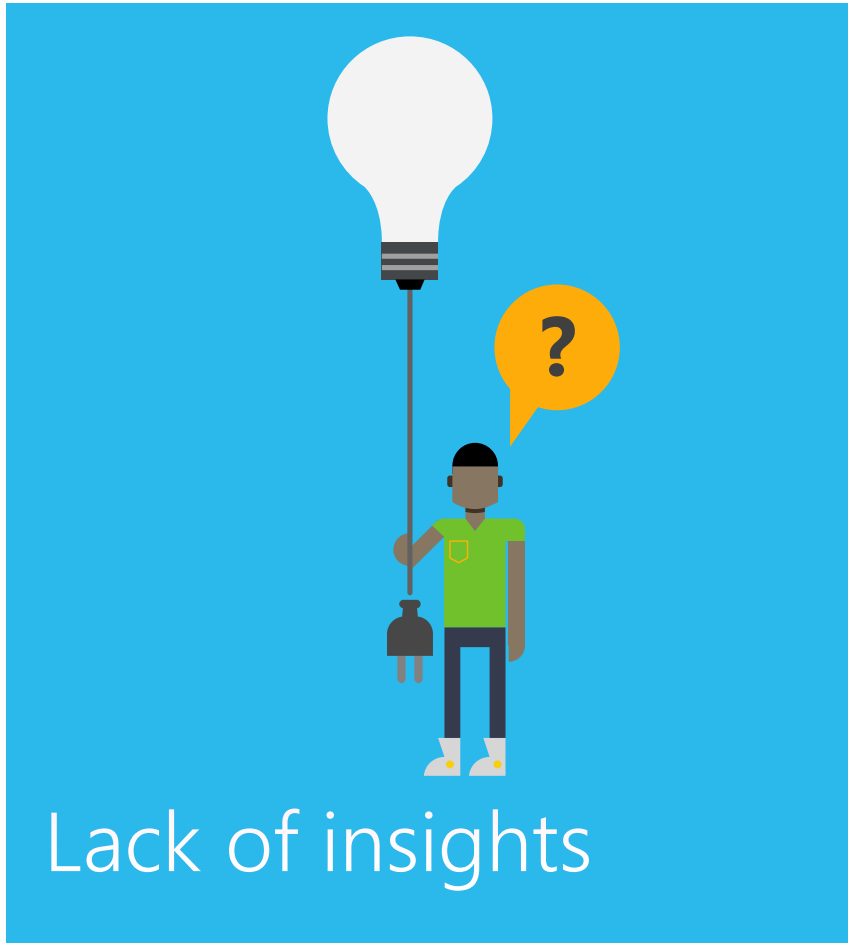
Traditional Development and Operations



Software delivery challenges



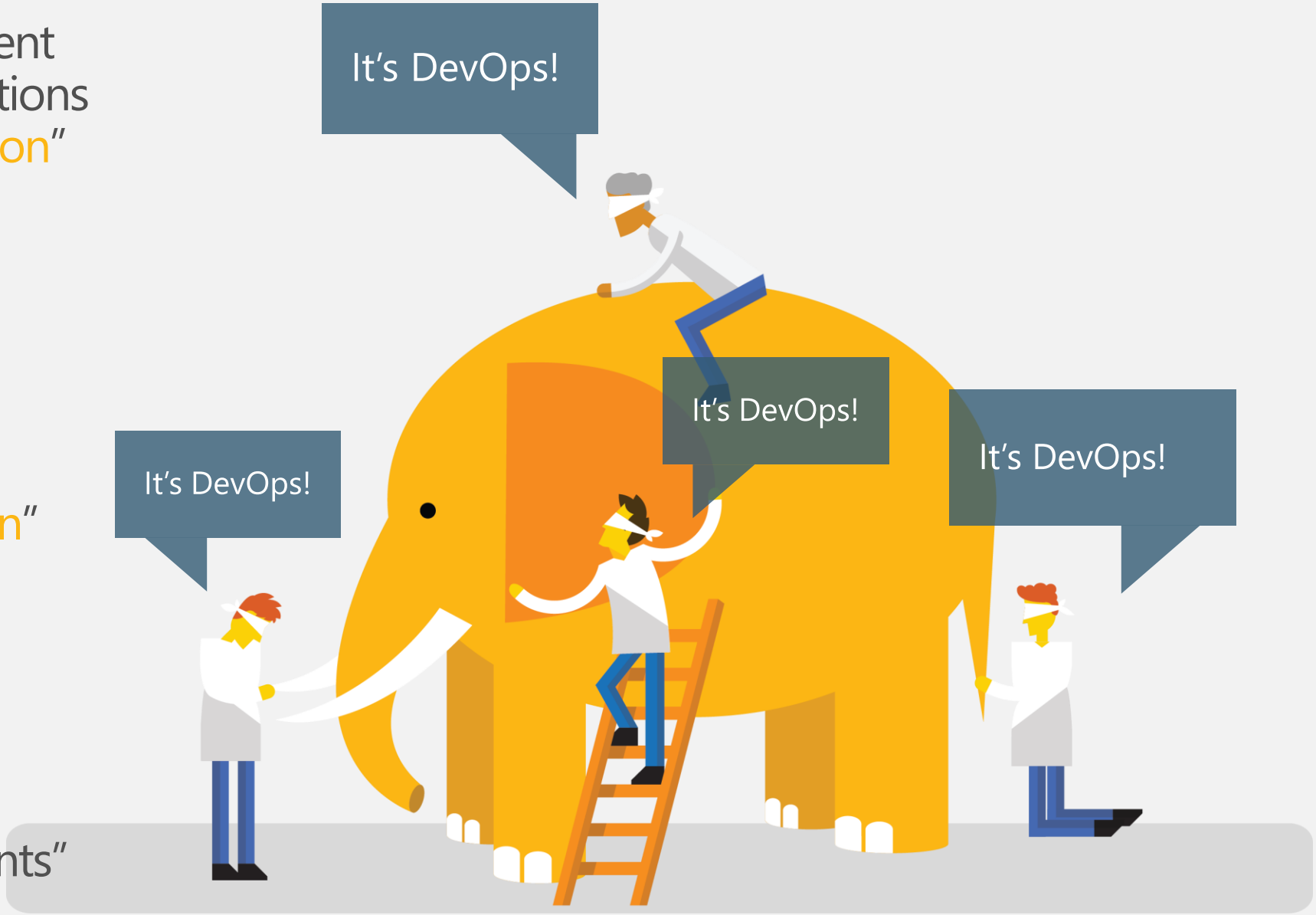




"**DevOps** is development and operations collaboration"

"**DevOps** is using automation"

"**DevOps** is small deployments"

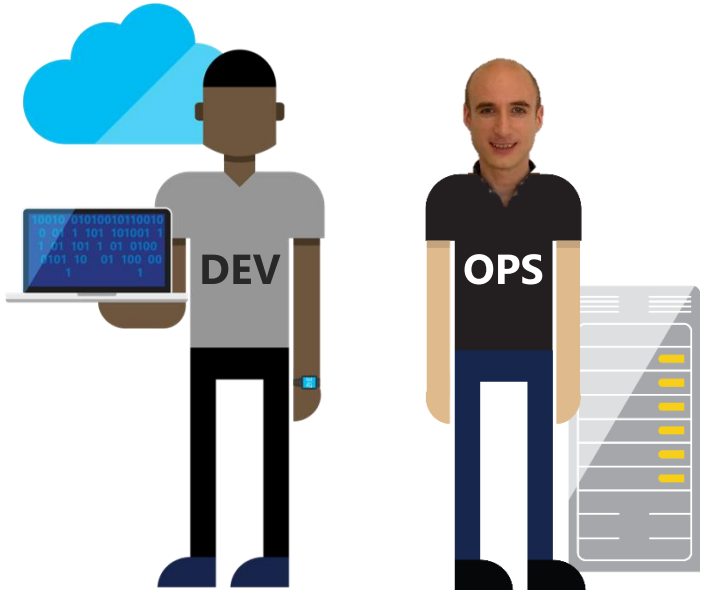


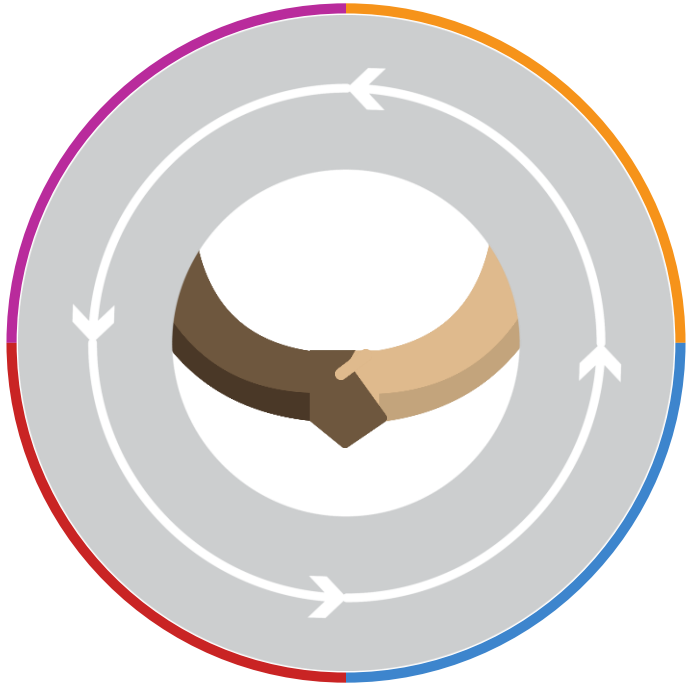
"**DevOps** is treating your infrastructure as code"

"**DevOps** is feature switches"

"Kanban for Ops?"

DevOps: the three stage conversation





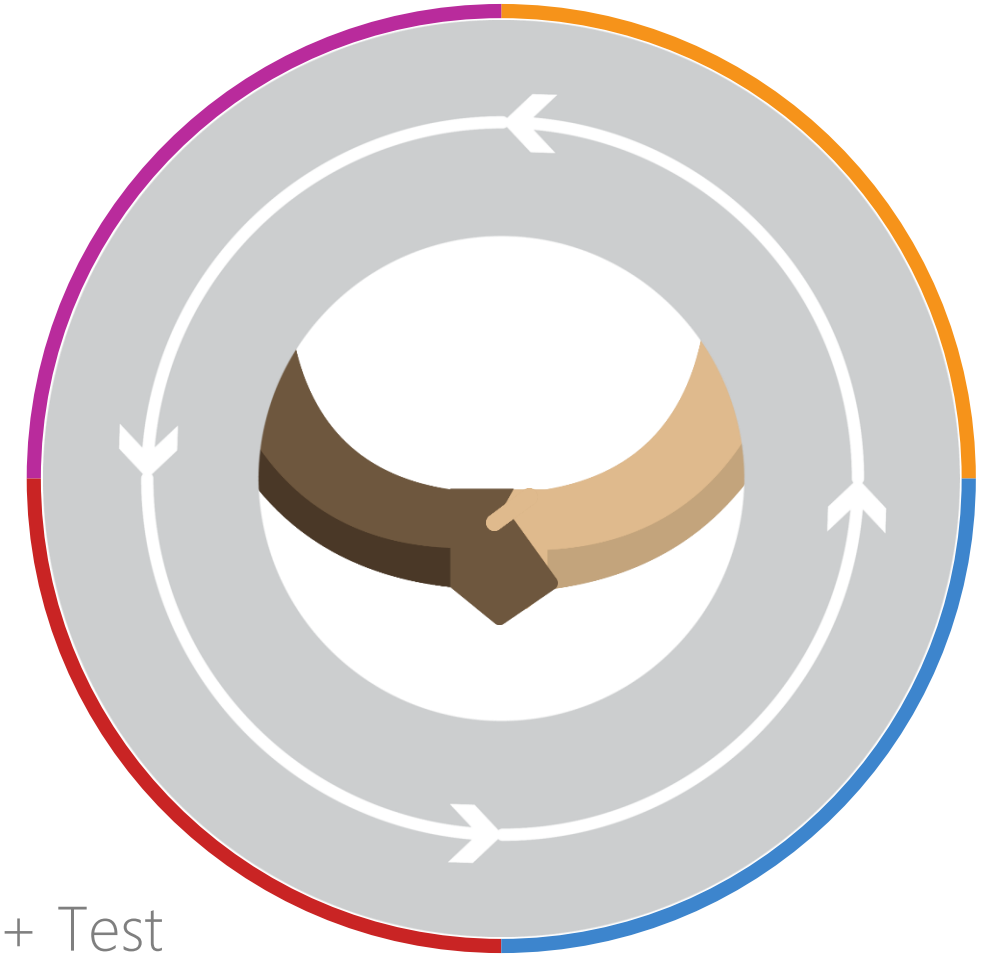
1	People
2	Process
3	Products

DevOps

1 Plan

4 Monitor + Learn

Development

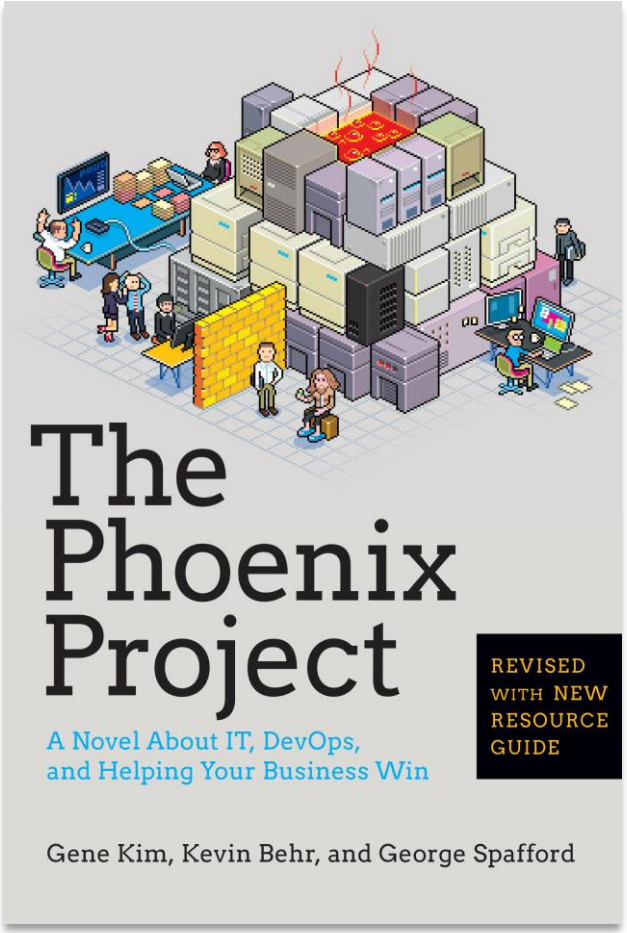


Production

2 Develop + Test

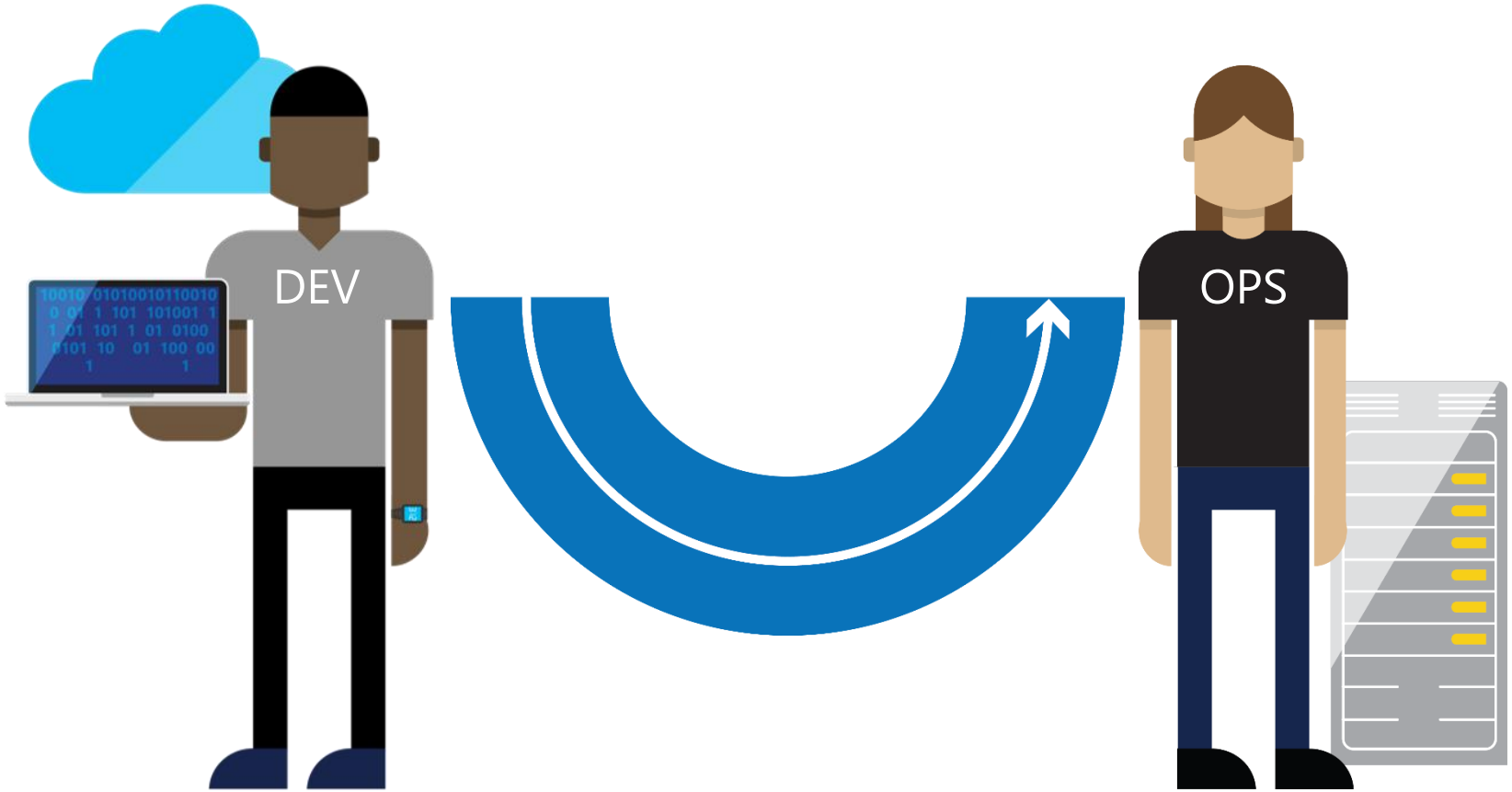
3 Release

The Three Ways

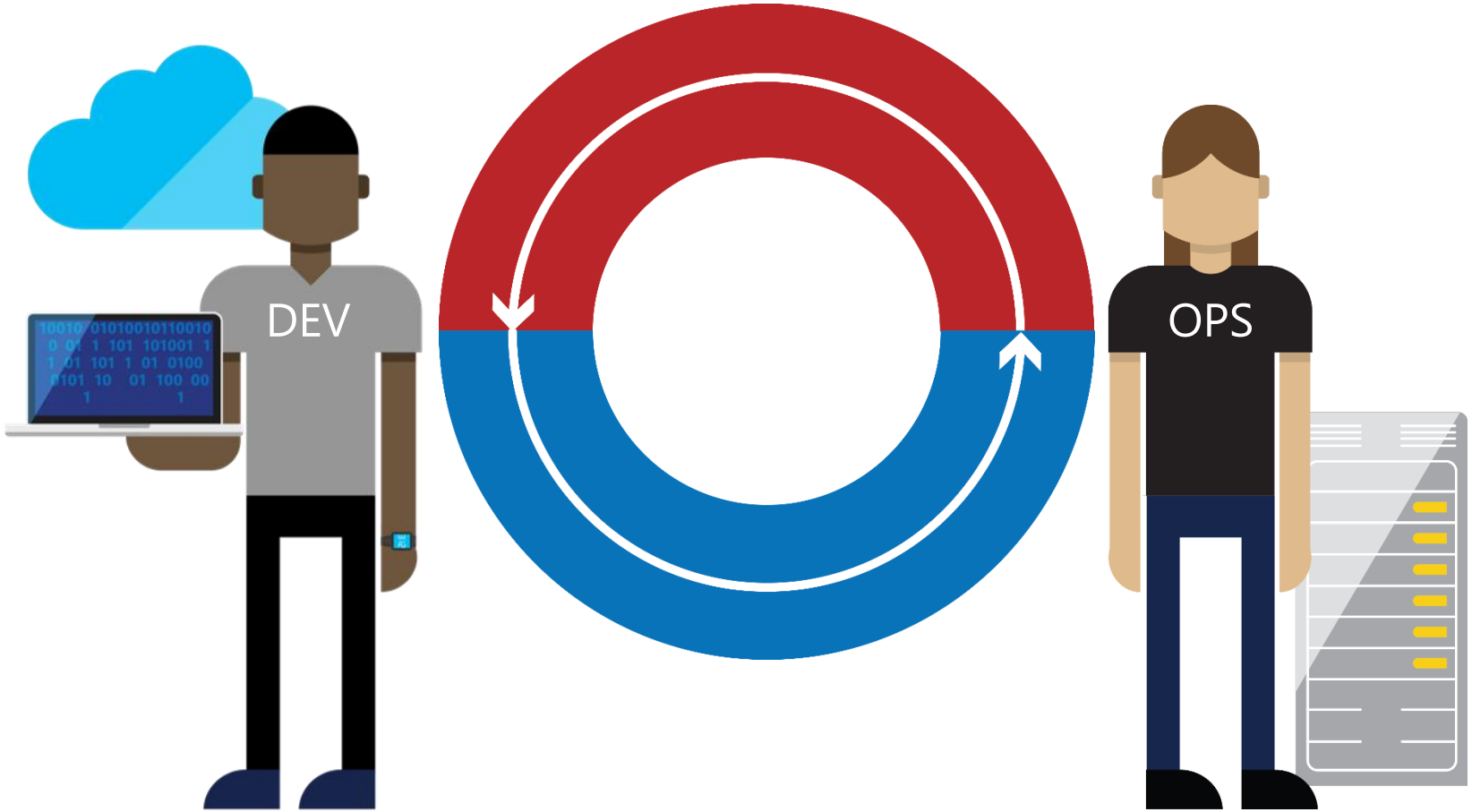


http://itrevolution.com/the-three-ways-principles-underpinning-devops/

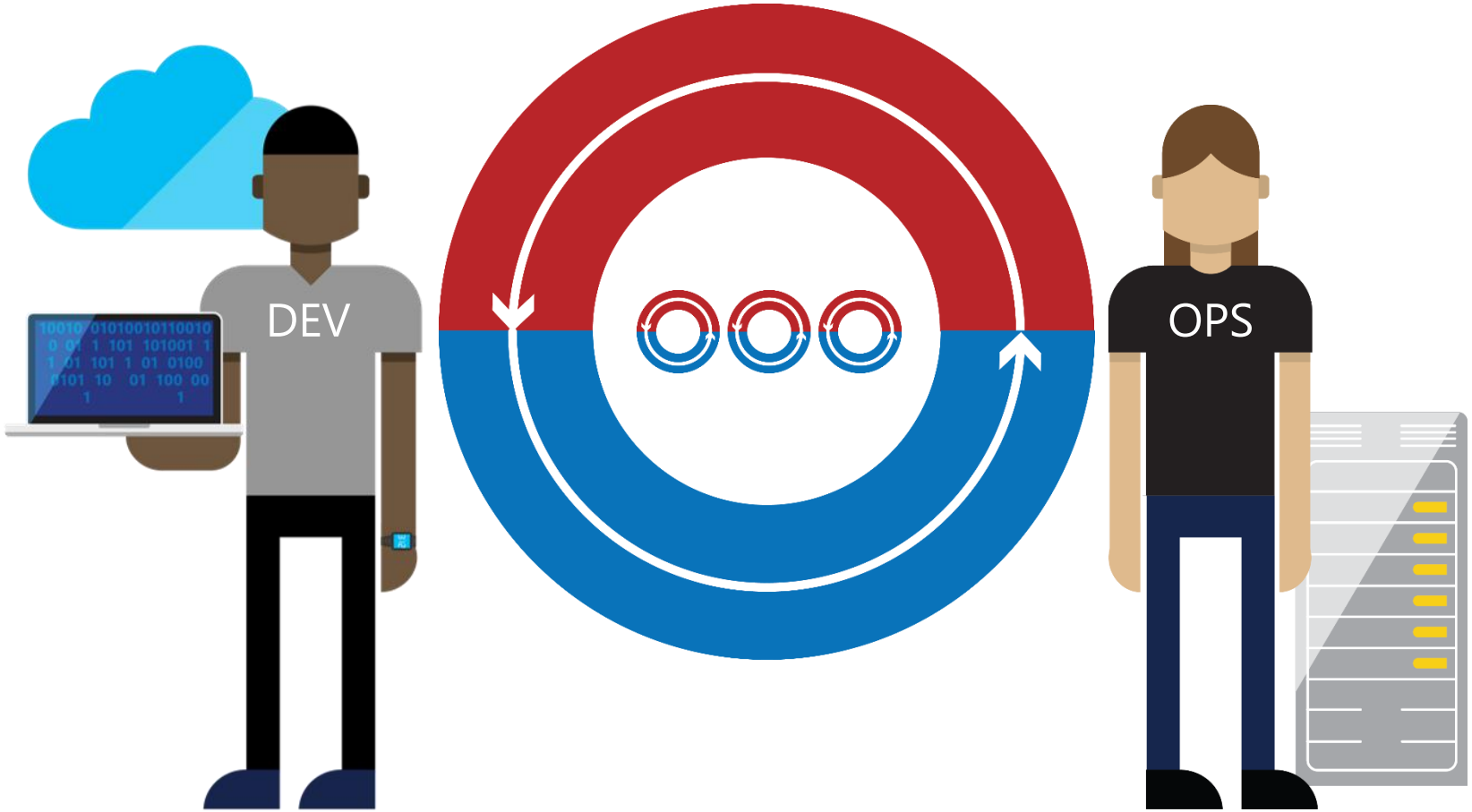
The first way (systems thinking)



The second way (feedback loops)



The third way (experimentation)



Containers Overview

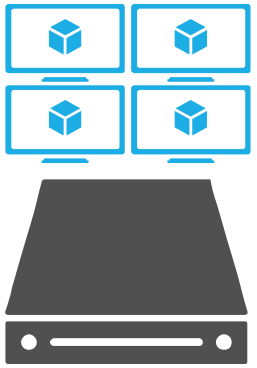
Containers

A new approach to build, ship, deploy, and instantiate applications

Physical

Applications traditionally built and deployed onto physical systems with 1:1 relationship

New applications often required new physical systems for isolation of resources



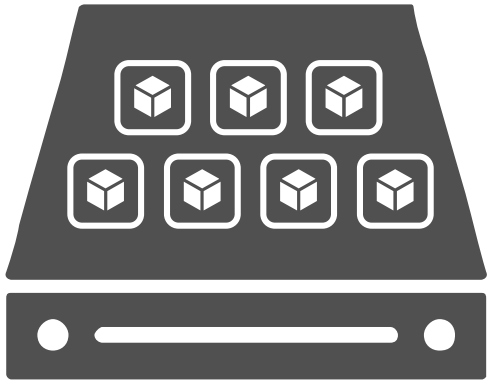
Virtual

Higher consolidation ratios and better utilization

Faster app deployment than in a traditional, physical environment

Apps deployed into VMs with high compatibility success

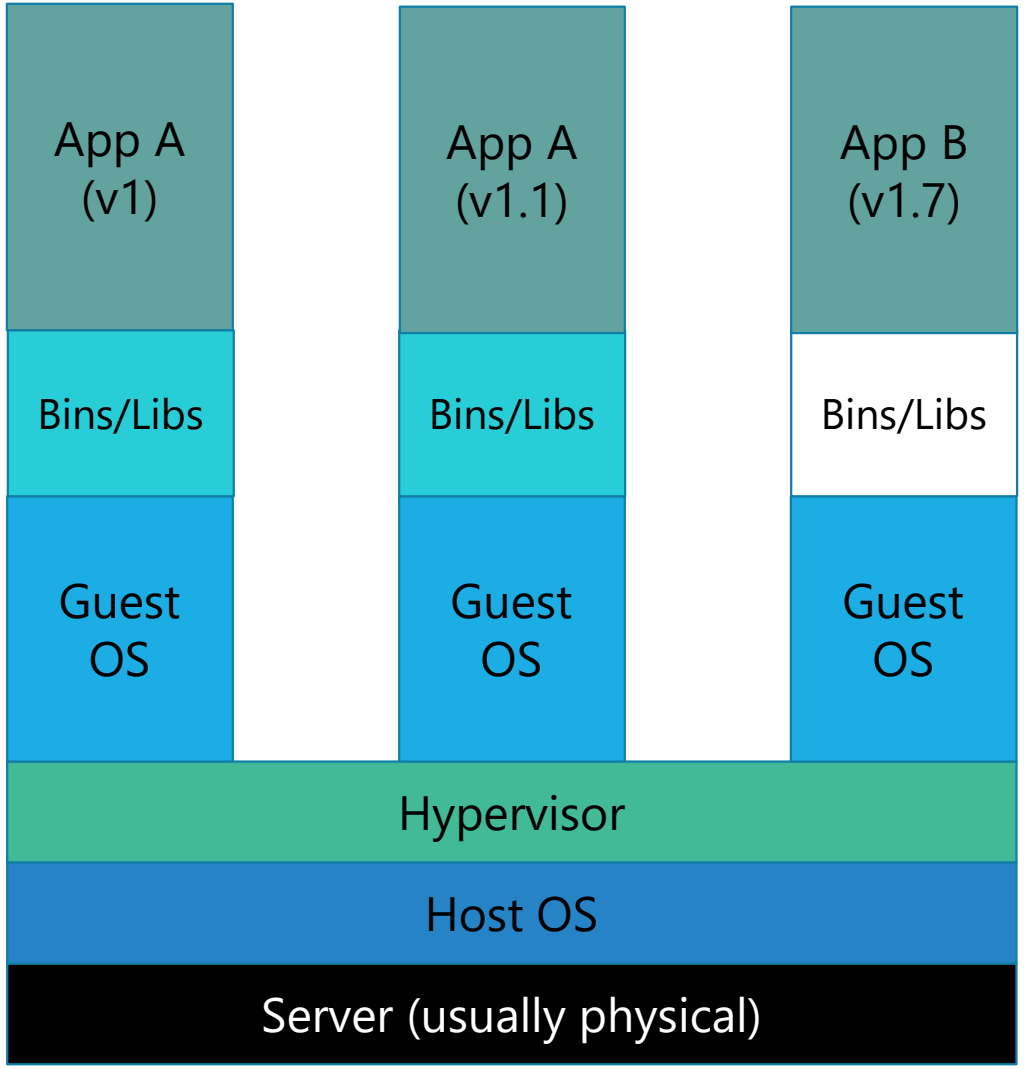
Apps benefited from key VM features i.e., live migration, HA



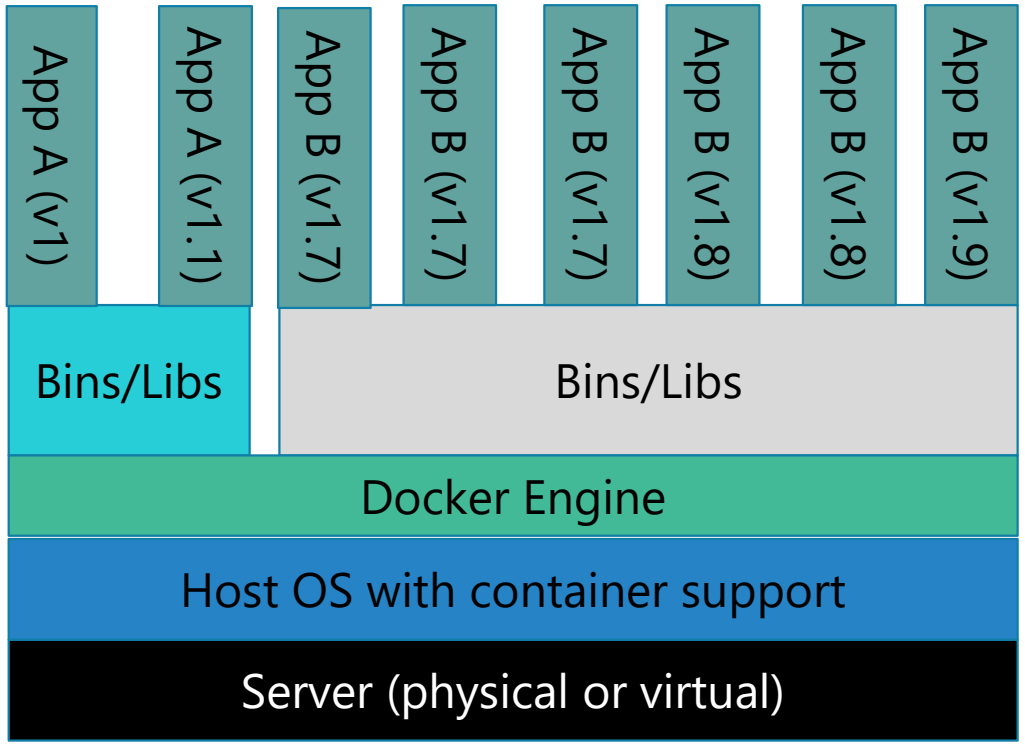
Containers
(either physical or virtual)

Package and run apps within containers

Server Virtualisation: Each app and each version of an app has dedicated OS



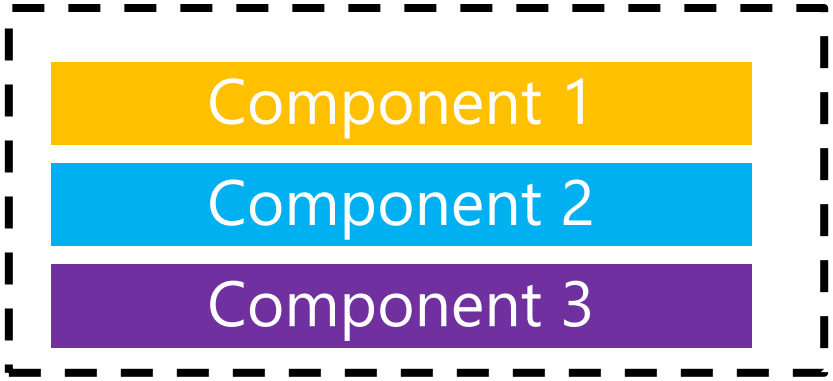
Containers: All containers share host OS kernel and appropriate bins/libraries



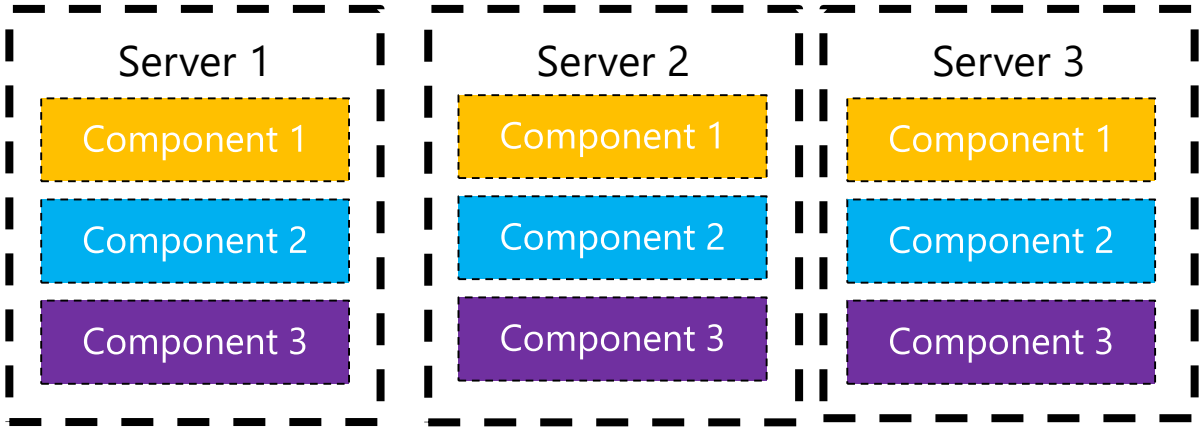
Container Facts

- Increase server consolidation
- Fast Start
  - Operating system is already running
- Portable
  - Reduces the “it works on my machine” problem
  - Streamline development and testing
- Immutable
  - Short lifetime
  - Update the image, not the container
  - Data needs to be stored outside of the container

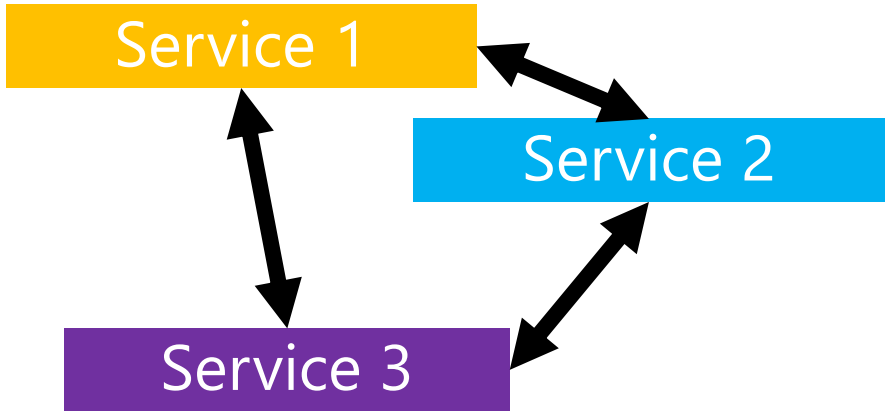
Monolithic Applications



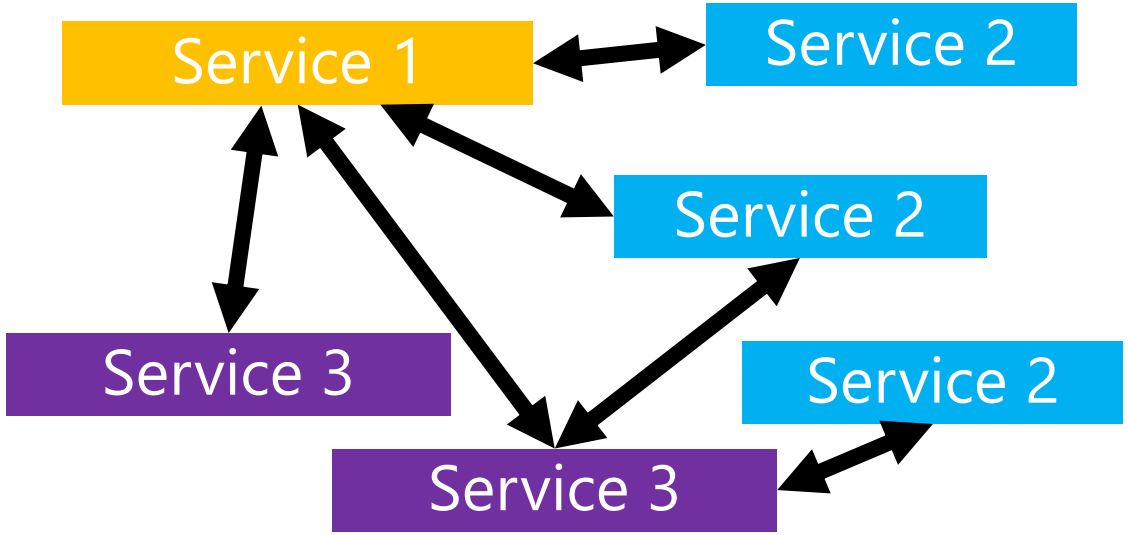
Scaled Monolith



Microservices



Scaled Microservices



Docker

Docker integration

Docker

An open source engine that automates the deployment of any application as a portable, self-sufficient container that can run almost anywhere

Partnership

Enable the Docker client to manage multi-container applications using both Linux and Windows containers, regardless of the hosting environment or cloud provider

Docker Hub

Huge collection of open and curated applications available for download. https://hub.docker.com

Collaboration

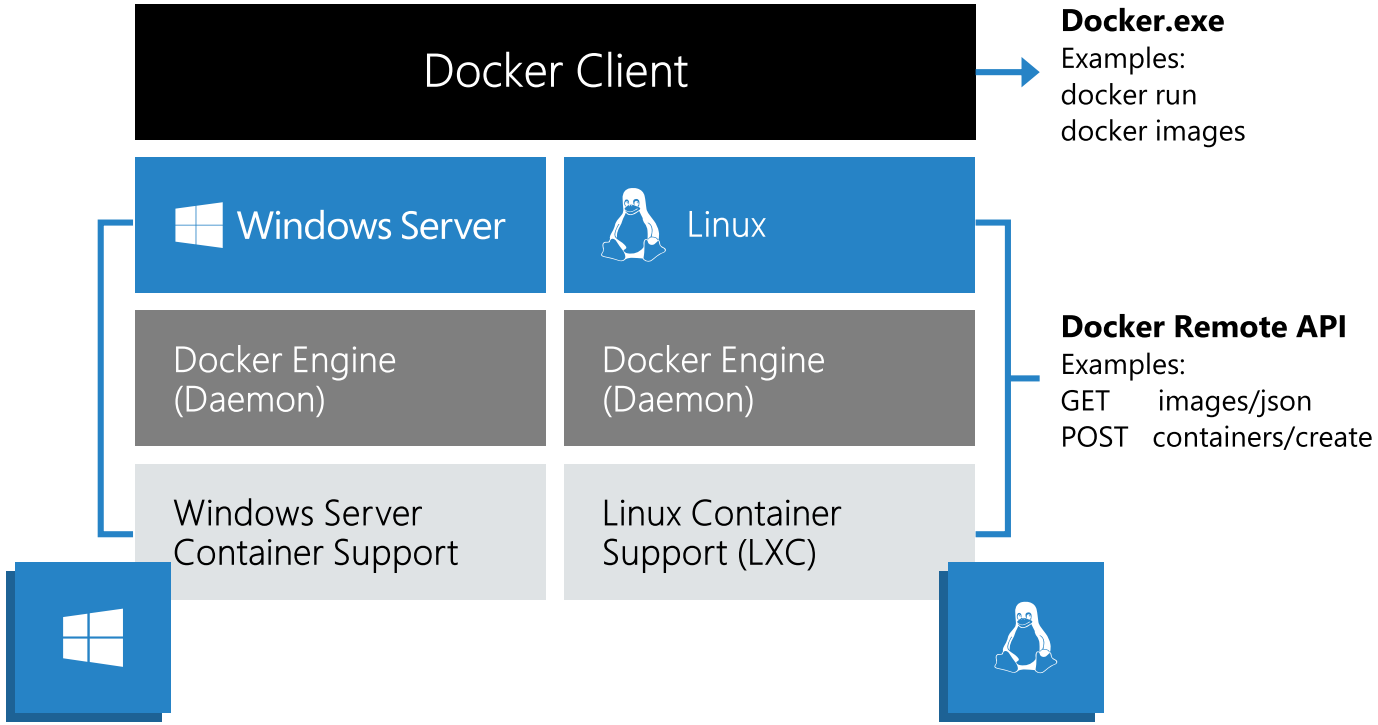
Bring Windows Server containers to the Docker ecosystem to expand the reach of both developer communities

Docker Engine

Docker Engine for Windows Server containers is developed under the Docker open source project

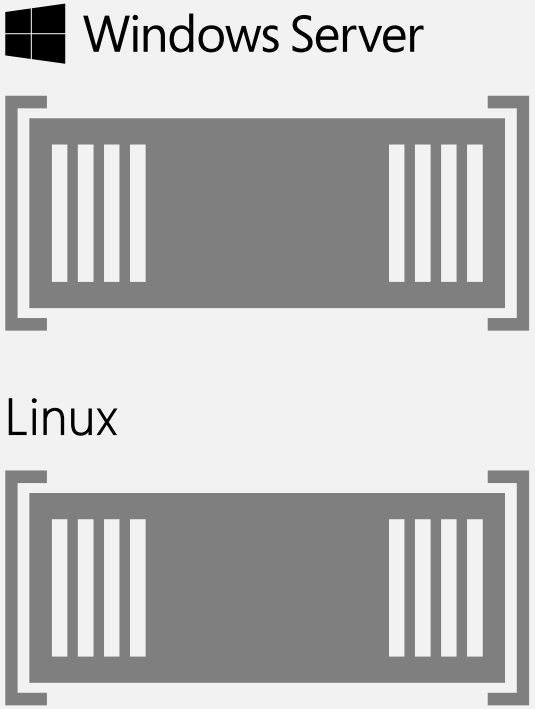
Docker Client

Windows customers can use the same standard Docker client and interface on multiple development environments



Container ecosystem

Container runtime



Container images

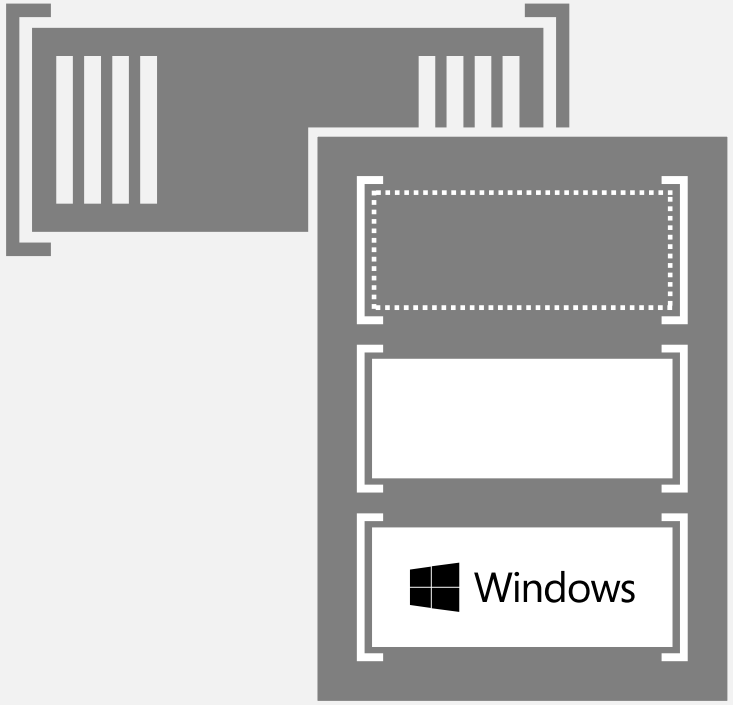
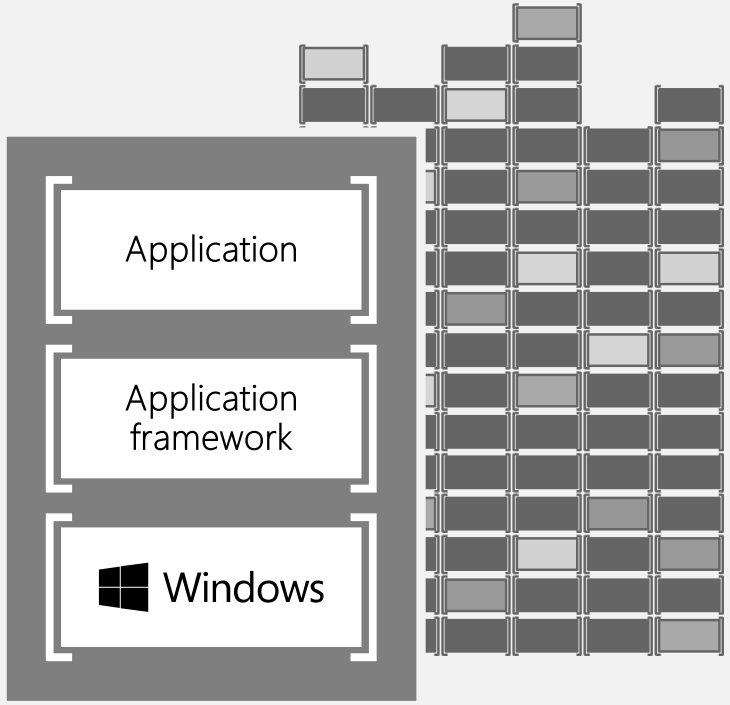


Image repository



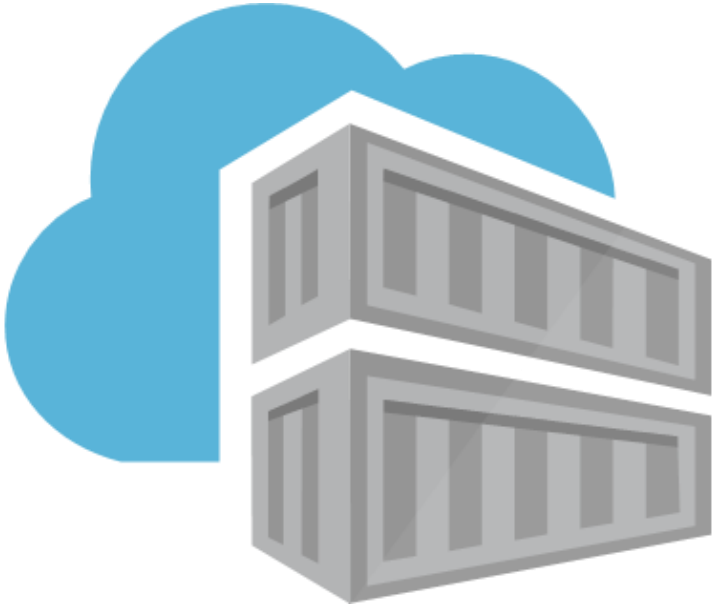
Container Images

- When containers are run a container image is run
- An image can be created manually by modifying a running container
- An image can be defined within a Dockerfile and created using **`docker build`**

```
# Indicates that the windowsservercore image will be used as the base image.
FROM microsoft/windowsservercore

# Metadata indicating an image maintainer.
MAINTAINER jshelton@contoso.com

# Uses dism.exe to install the IIS role.
RUN dism.exe /online /enable-feature /all /featurename:iis-webserver /NoRestart

# Creates an HTML file and adds content to this file.
RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html

# Sets a command or process that will run each time a container is run from the new image.
CMD [ "cmd" ]
```

Container Registry

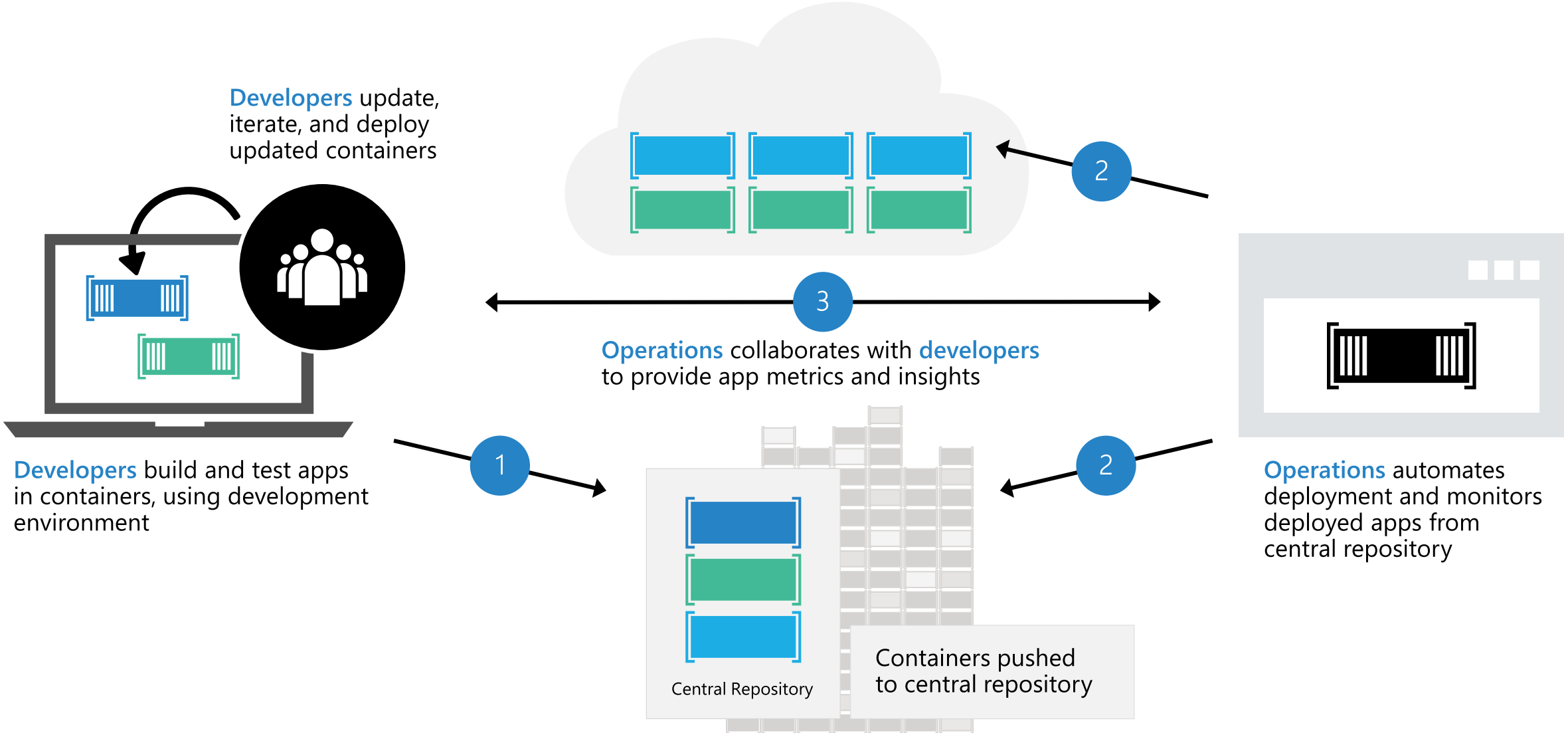
- A place to store and distribute images
- Examples of registries include:
  - Docker Hub – hosted by Docker
  - Private Registry
  - Azure Container Registry
- Create repository for each image
- Push image to repository using `docker push`



Running Containers

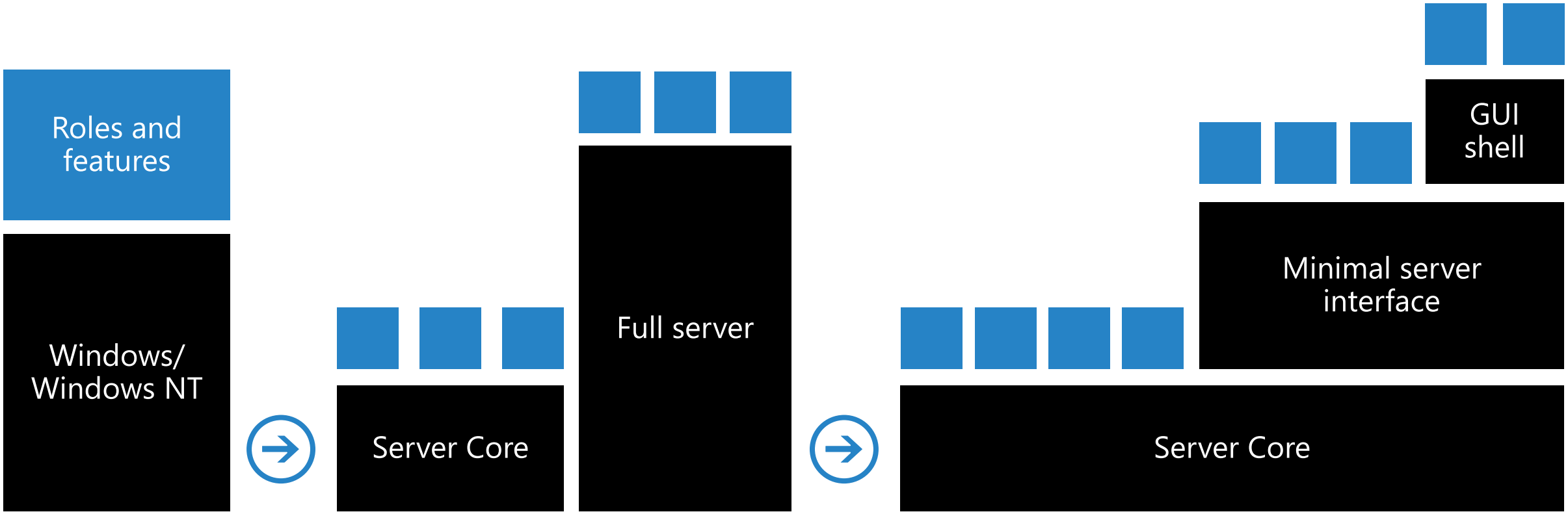
- Containers can be run on Linux server with Docker Engine installed
- Run a container image using **`docker run`**

DevOps process with containers



Windows Server Containers

The story so far…



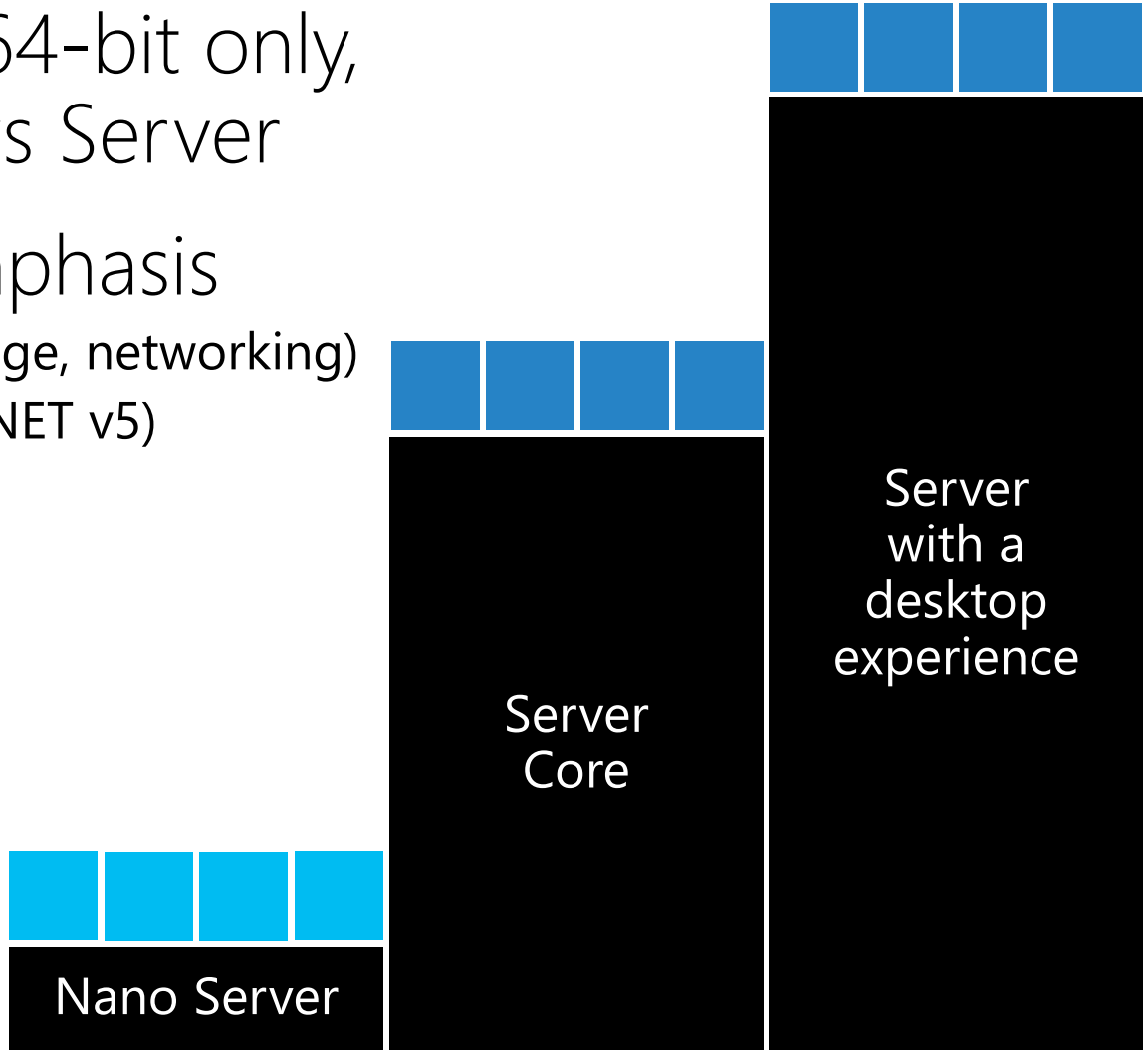
Windows NT to Windows Server 2003

Windows Server 2008 and Windows Server 2008 R2

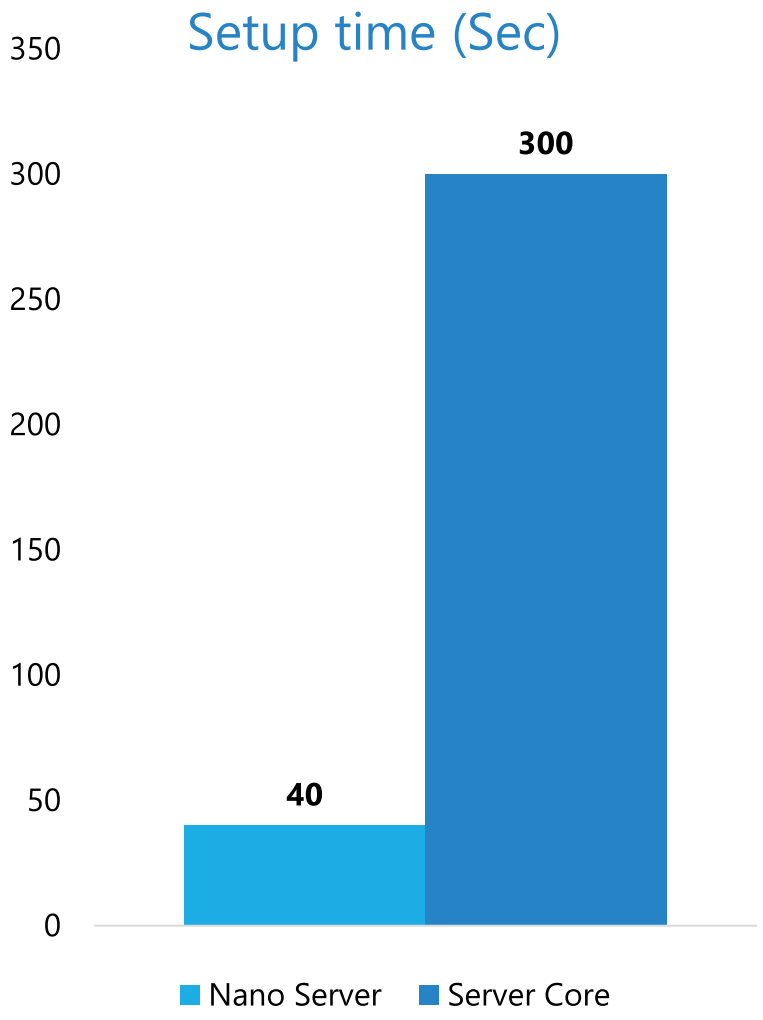
Windows Server 2012 and Windows Server 2012 R2

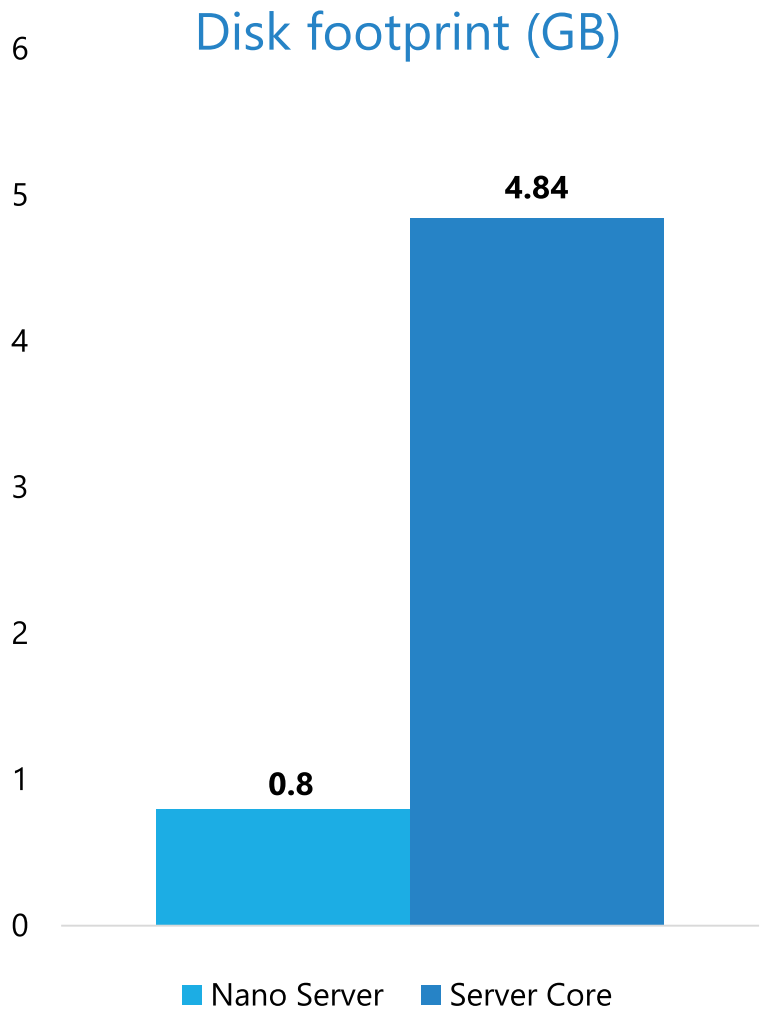
The next step in the journey…

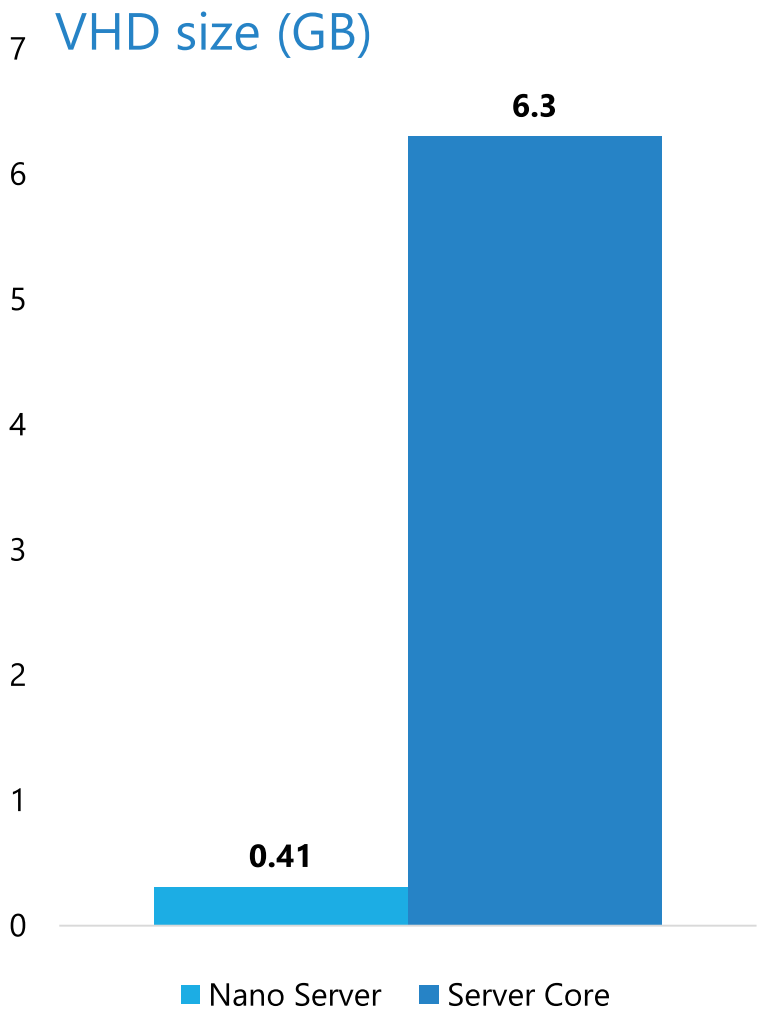
- **Nano Server**: A new headless, 64-bit only, deployment option for Windows Server
- Deep refactoring with cloud emphasis
  - Cloud fabric & infrastructure (clustering, storage, networking)
  - Born-in-the-cloud applications (PaaS v2, ASP.NET v5)
  - VMs & Containers (Hyper-V & Docker)
- Extend the Server Core pattern
  - Roles & features live outside of Nano Server
  - No binaries or metadata in OS image
  - Standalone packages install like apps
  - Full driver support
  - Antimalware



Deployment improvements







Windows Server Container OS environments

Nano Server

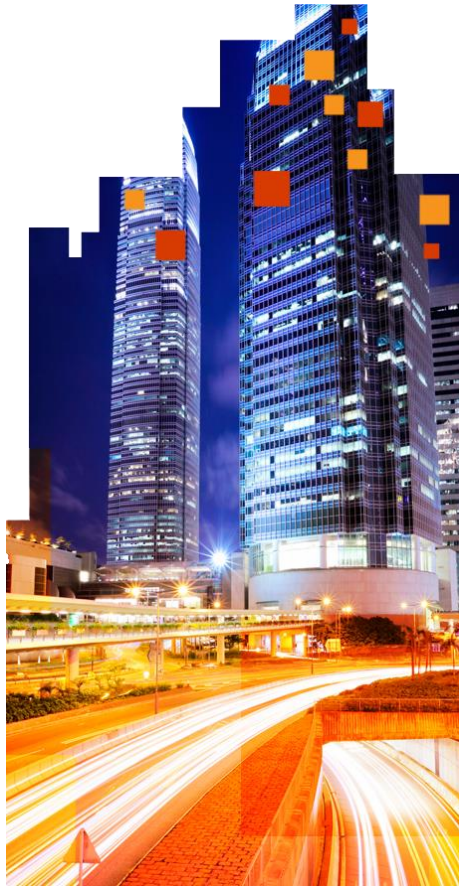
Highly optimized

"Born in the cloud" applications

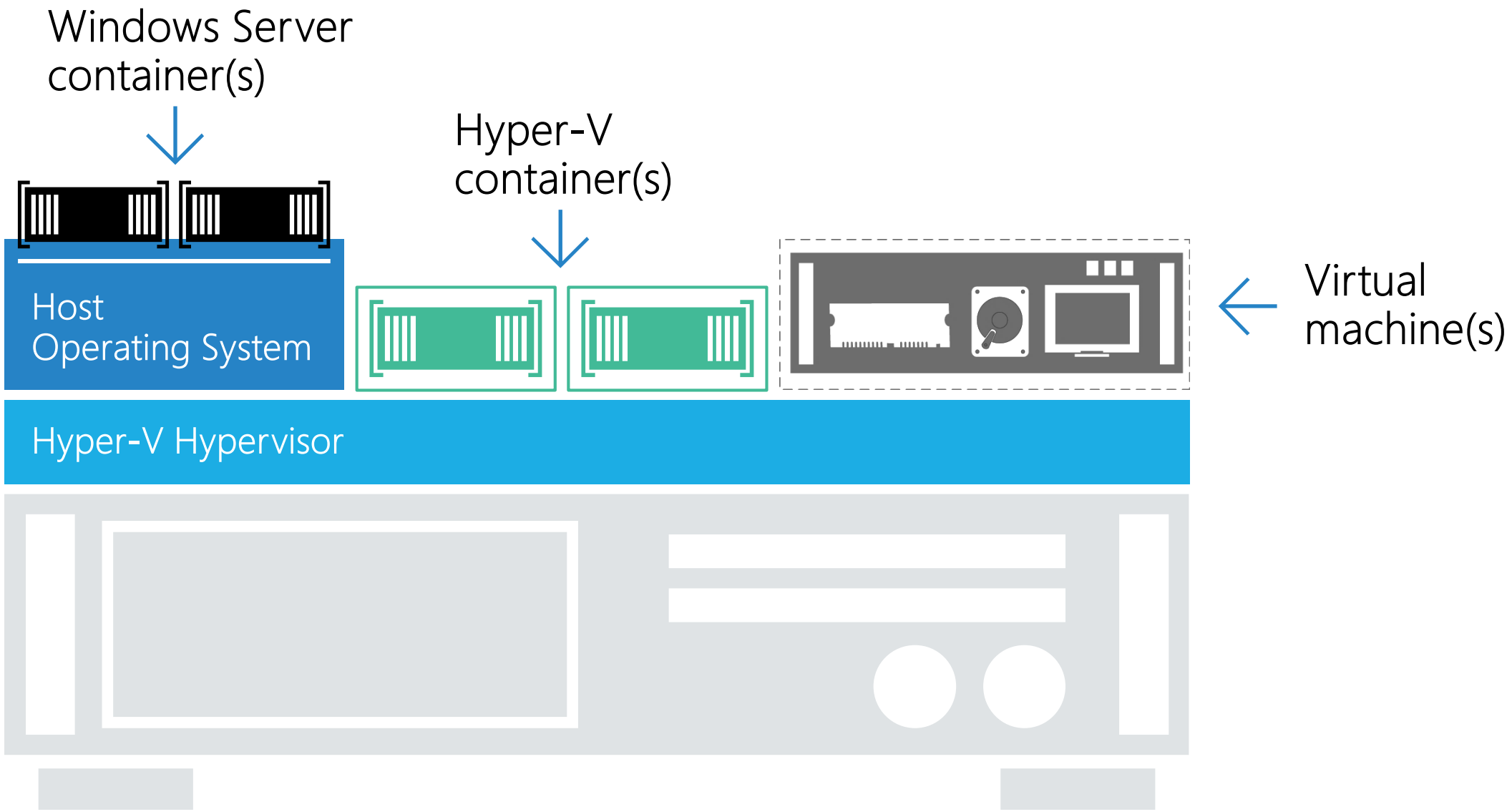
Server Core

Highly compatible

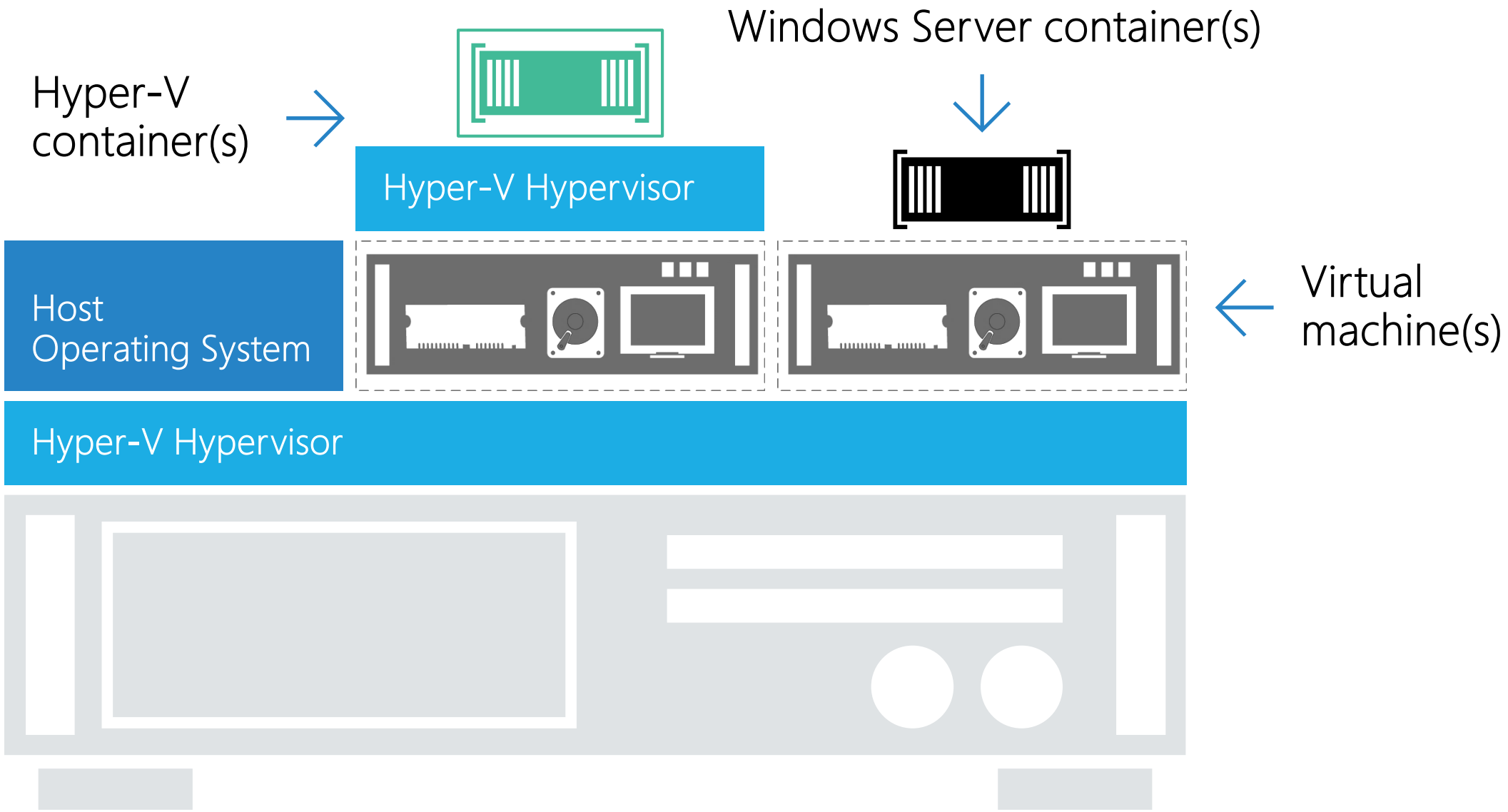
Traditional applications



Hyper-V Containers



Hyper-V Containers using nested virtualization



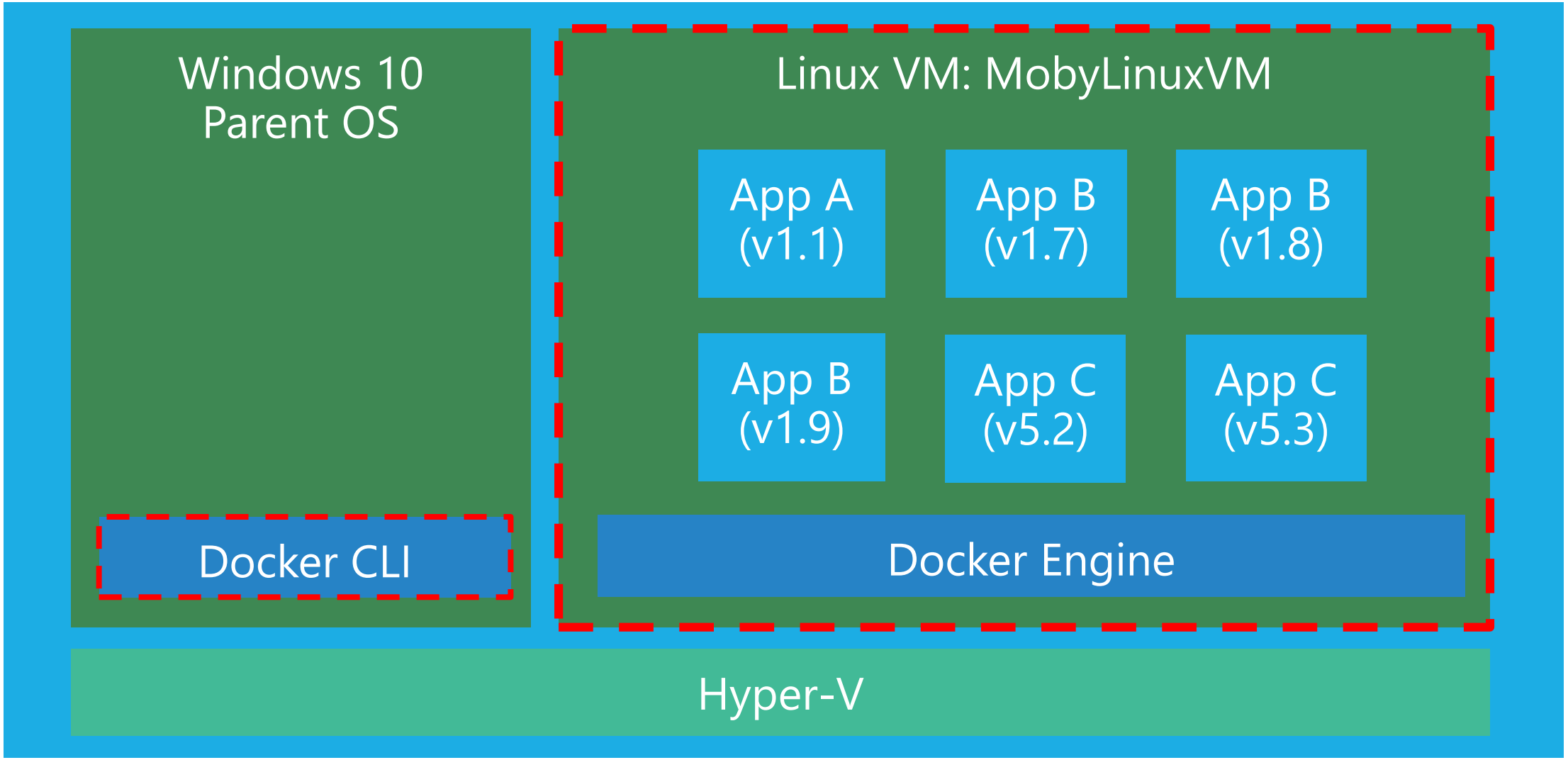
DEMO

Windows Server Containers

Docker Community Edition

(formerly Docker for Windows)

Docker for Windows

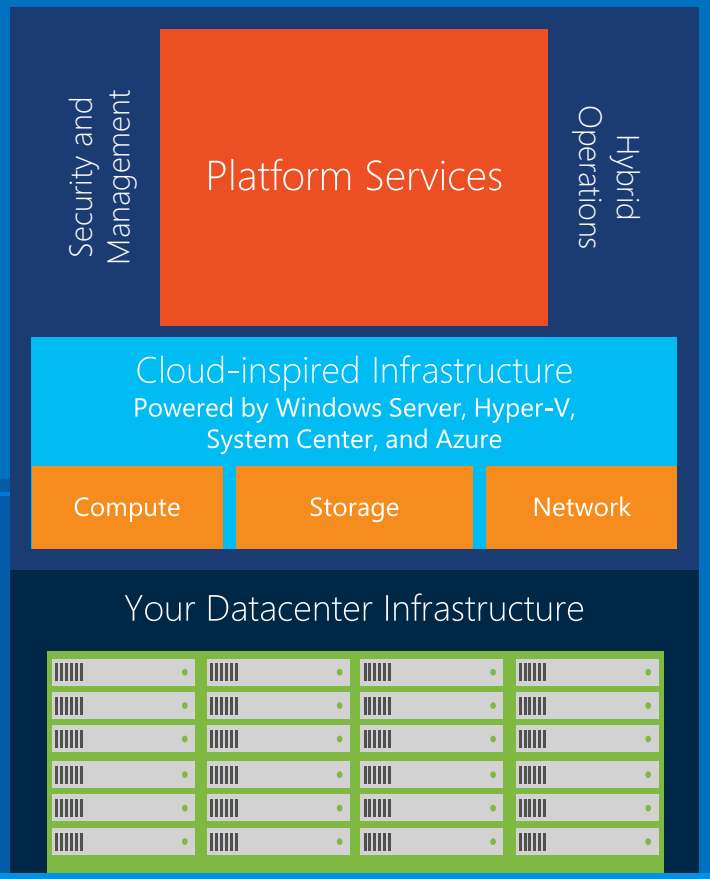


DEMO

Docker Community Edition

Azure and Containers

Microsoft Azure & Azure Stack



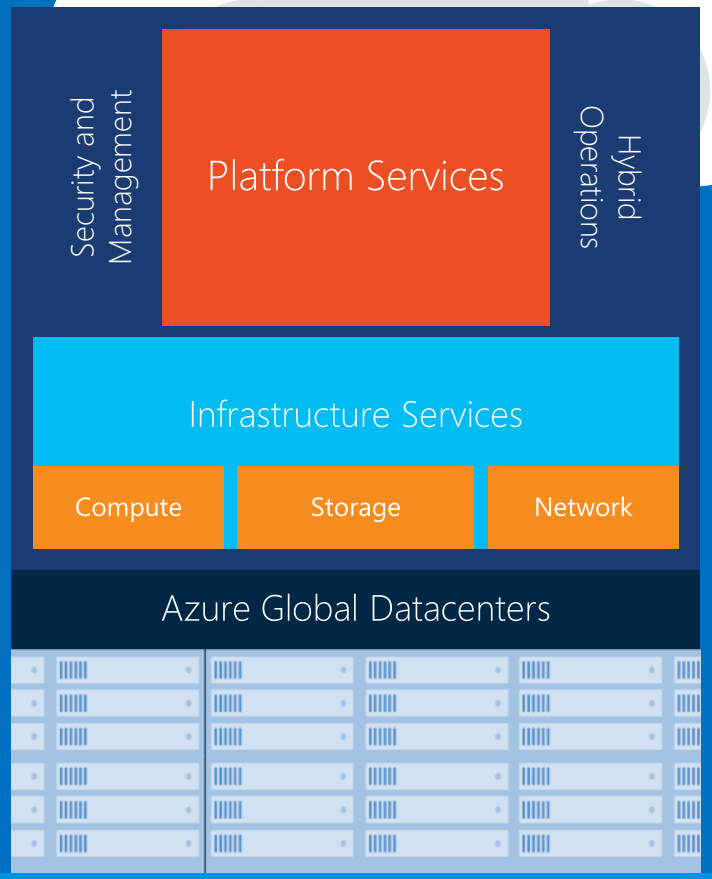
Developers

One Azure ecosystem

Unified app development

Azure services in your datacenter

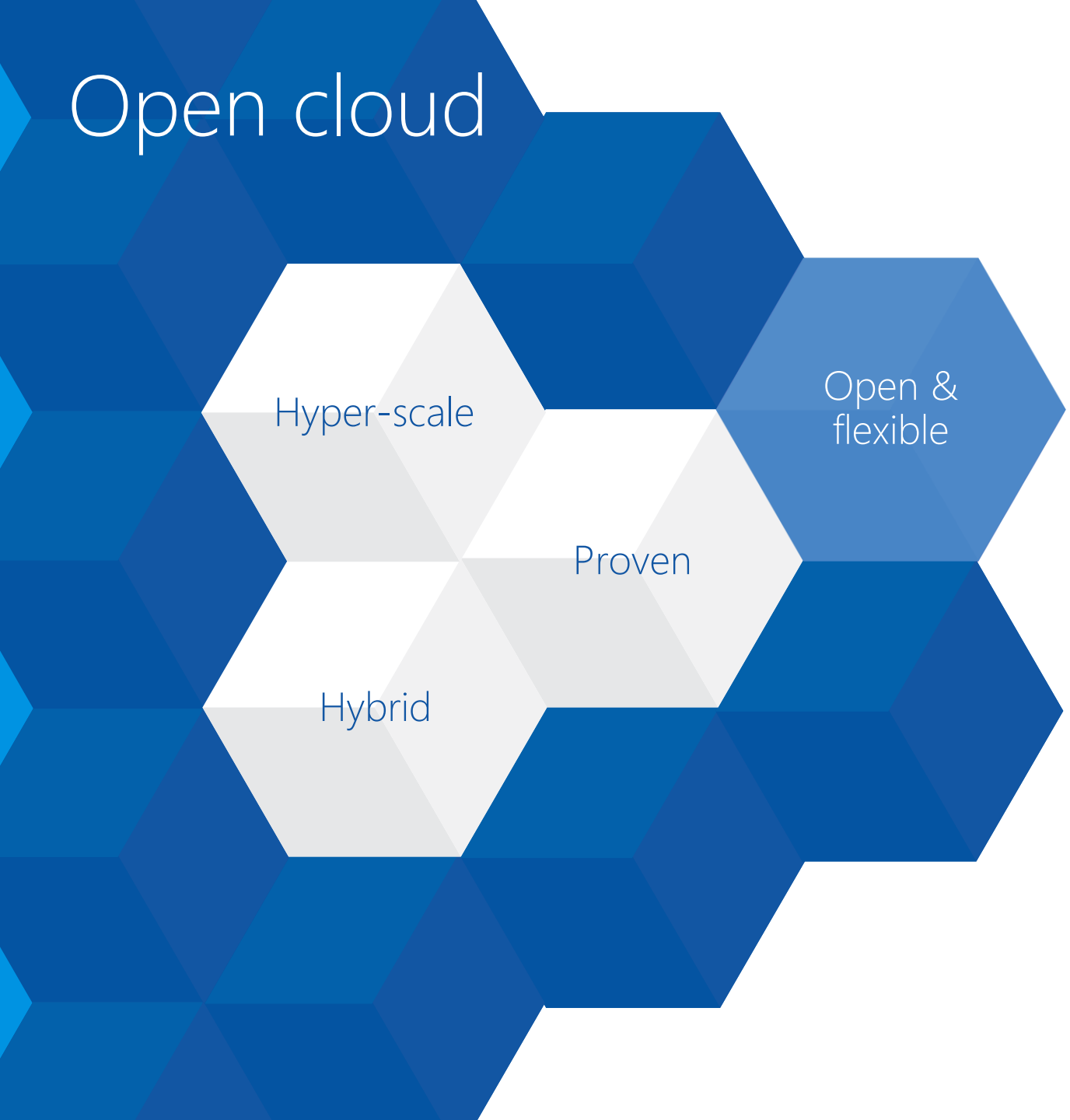
Operations



Microsoft Azure Stack
and Cloud Platform System

Microsoft Azure Cloud
Public, Global, Shared

Open cloud



Management

puppet labs, CHEF, ANSIBLE, Apache MESOS

Applications

apprenda®, Joomla!, Drupal, CLOUD FOUNDRY

App Frameworks

nodeJS, python, Java, php, Ruby

Databases & Middleware

cloudera, mongoDB, hadoop, MySQL

Infrastructure

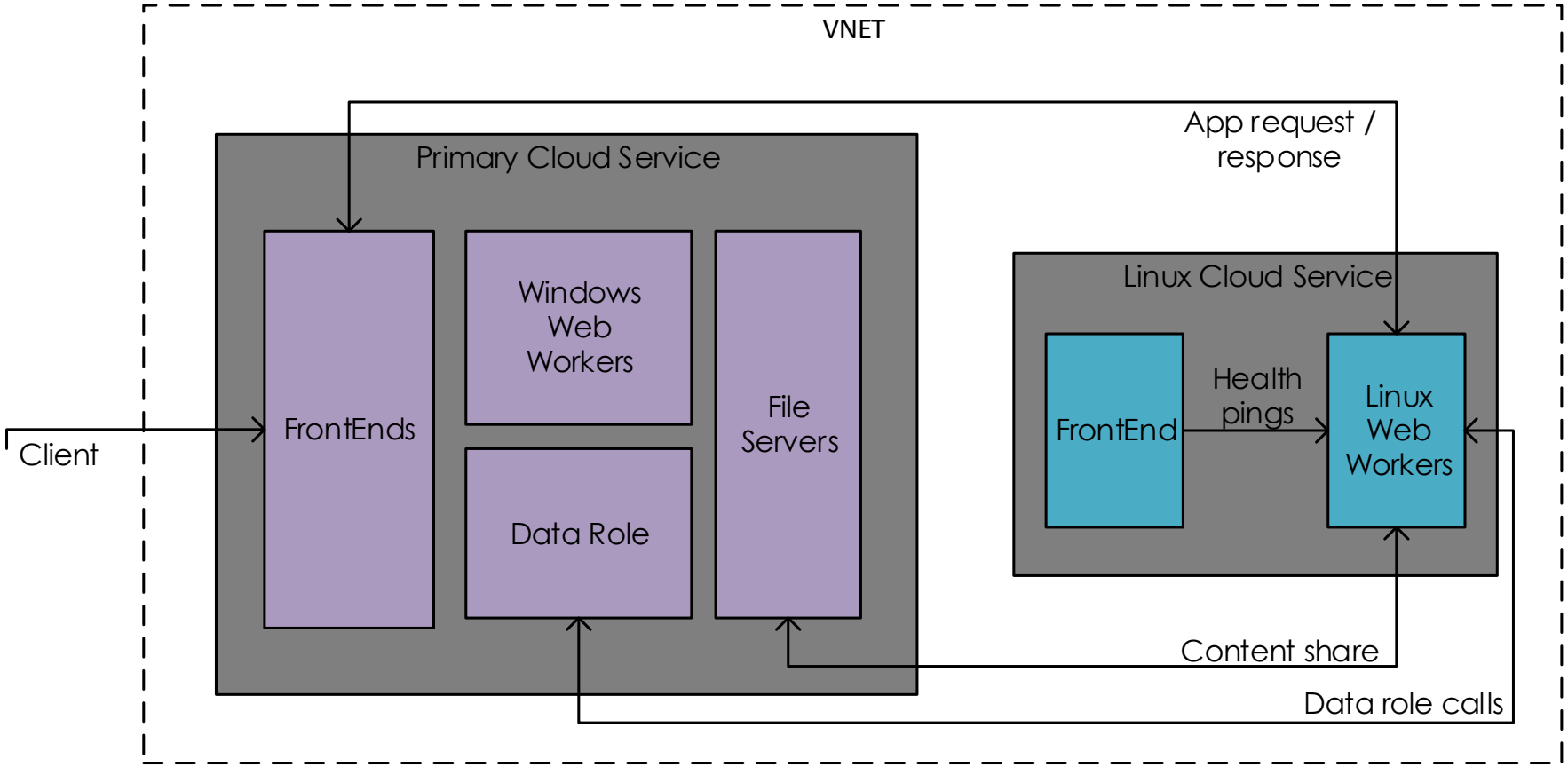
FreeBSD, Linux, docker, redhat

Containers on Linux on Azure App Service

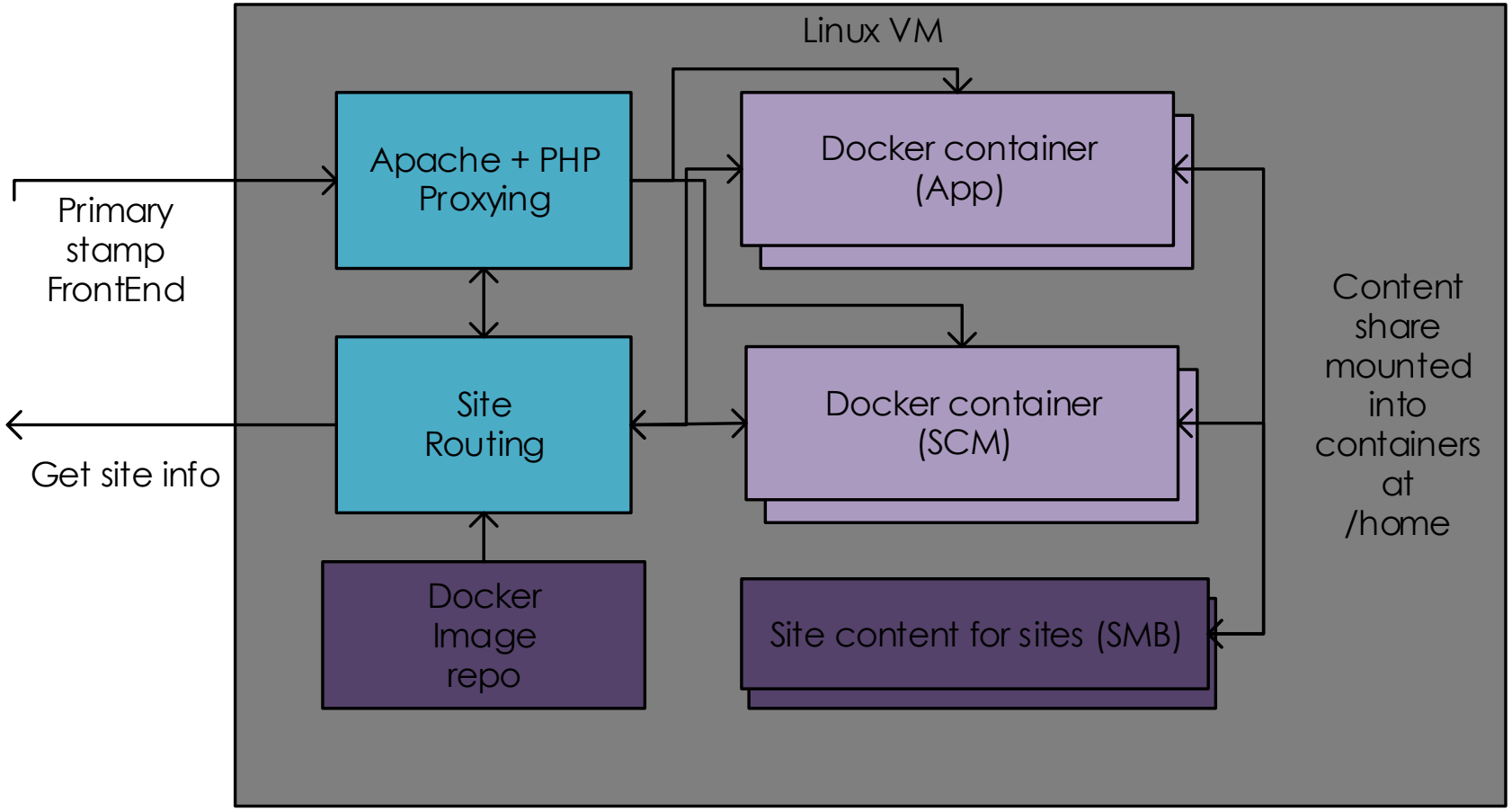
Create and deploy web sites natively in seconds

- Fastest way to build for the cloud
- Provision and deploy fast
- Run web apps natively on Linux
- Open and flexible for everyone
- Monitor, alert, and auto scale (preview)

Service Architecture



Linux Worker Architecture



marcusdemo - Docker Container
App Service

Search (Ctrl+/)

- Activity log
- Access control (IAM)
- Tags

APP DEPLOYMENT

- Quickstart
- Deployment credentials
- Deployment options

SETTINGS

- Application settings
- Docker Container
- Custom domains
- SSL certificates
- Scale up (App Service plan)
- Scale out (App Service plan)
- Push

Docker Container

Web Apps on Linux leverage the power of Docker containers to let you use custom containers from Azure Container Registry, Docker Hub, a private container registry, or use one of our default containers provided by App Service.

Image source

Built-in | Docker Hub | Private registry

* Image and optional tag (eg 'image:tag')

containerregistry-microsoft.azurecr.io/linuxwebsite:latest

* Server URL

https://containerregistry-microsoft.azurecr.io

* Login username

74824f88-020e-446b-ba3e-35f75f376987

* Password

••••••••

Startup Command

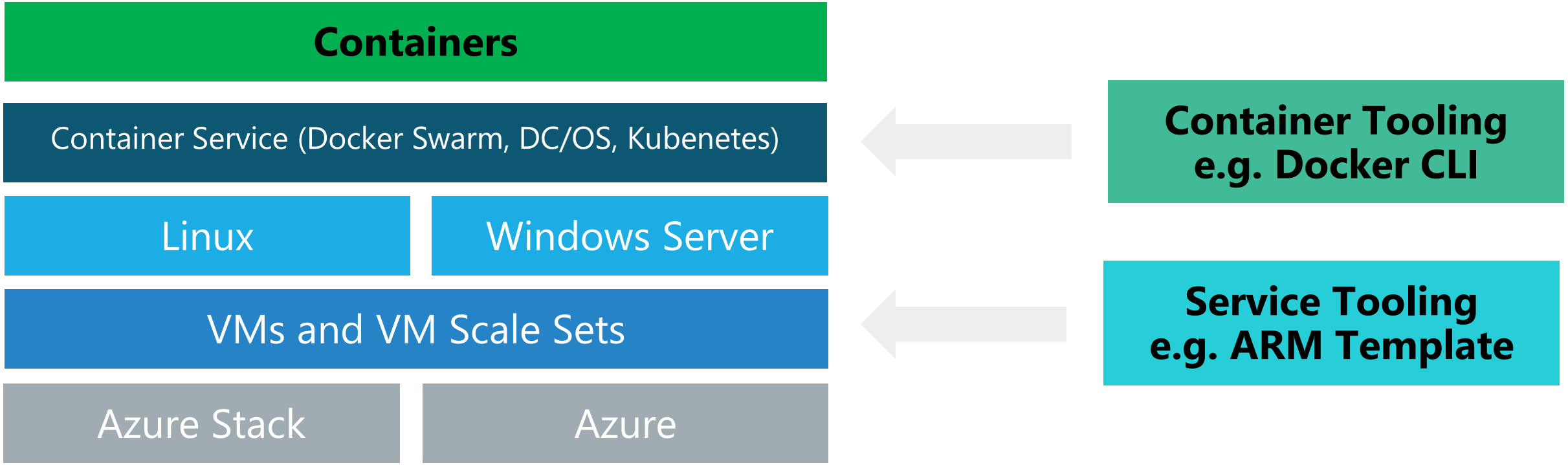
Save | Discard

DEMO

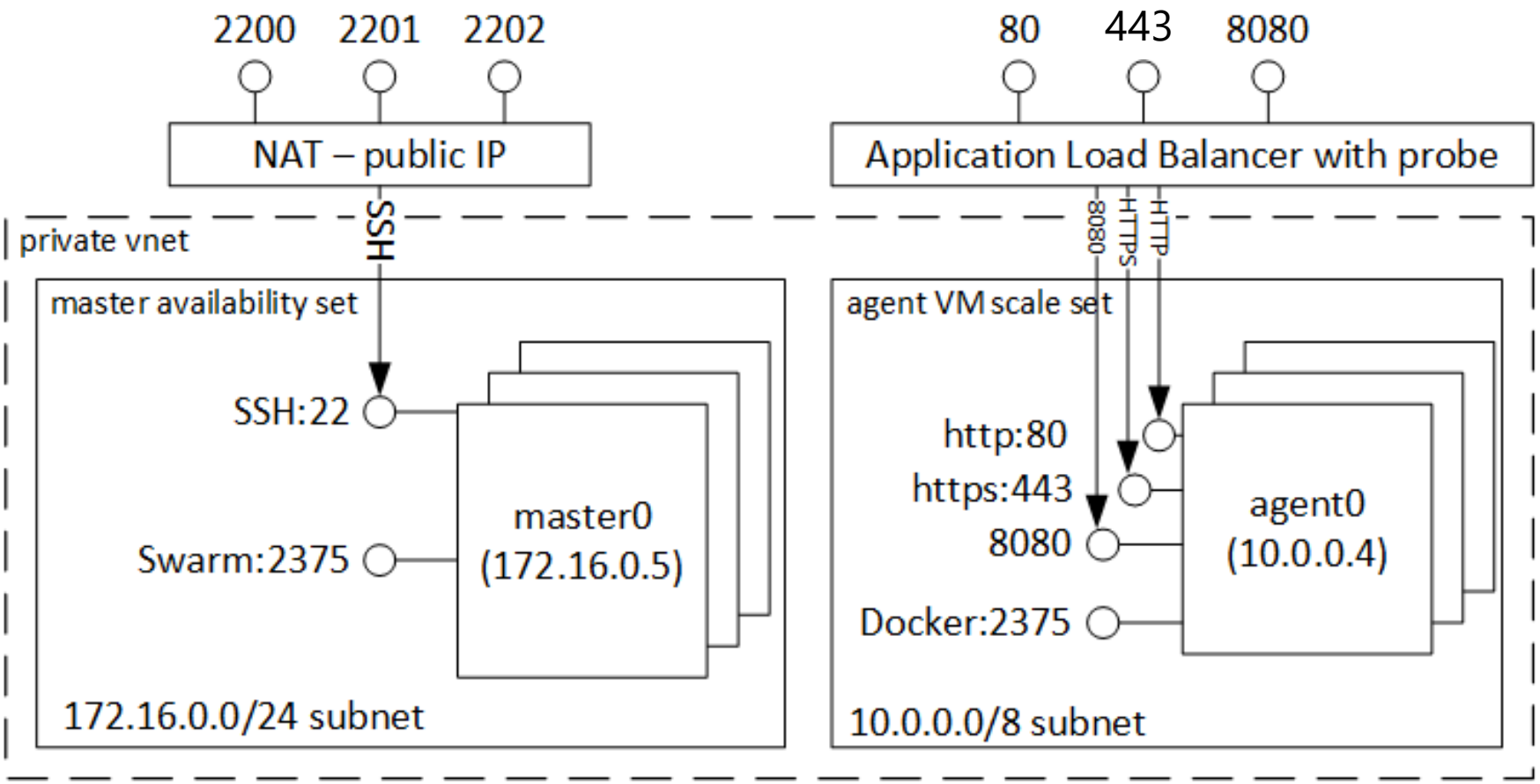
Docker
on Azure App Service

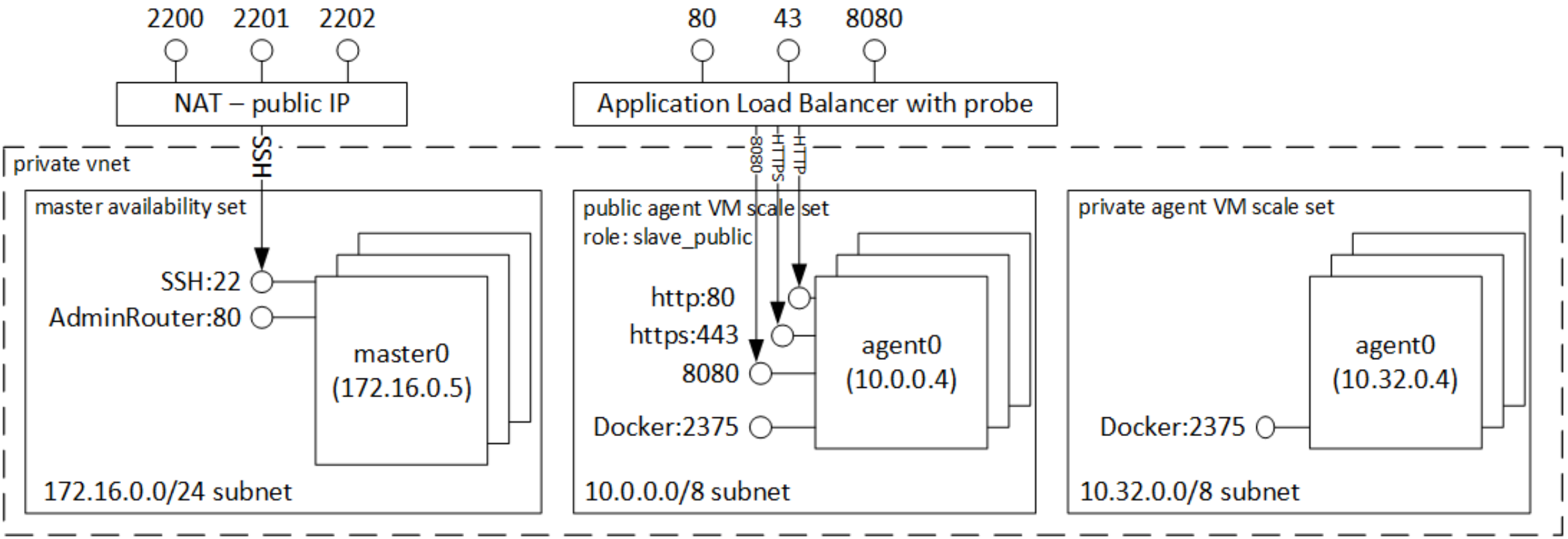
Azure Container Service

Azure Container Service

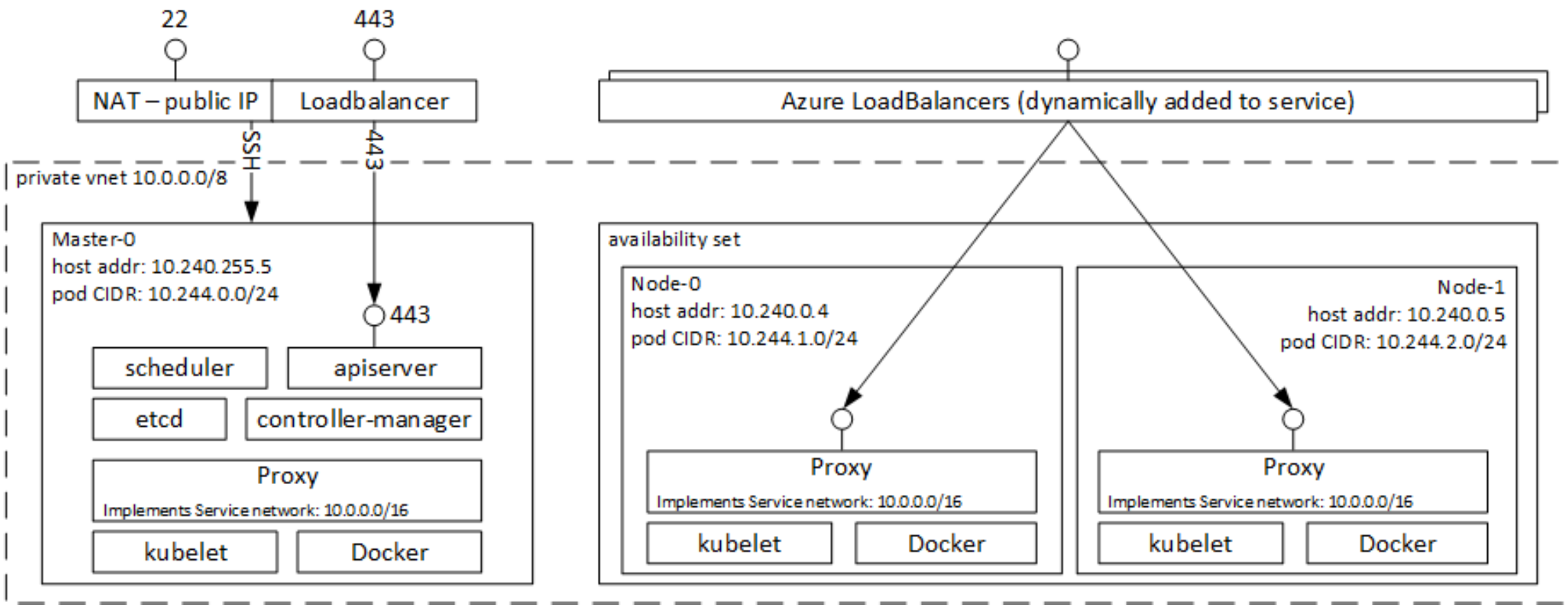


ACS with Swarm




2200
2201
2202
NAT – public IP
SSH
80
43
8080
Application Load Balancer with probe
8080
HTTPS
HTTP
private vnet
master availability set
SSH:22
AdminRouter:80
master0
(172.16.0.5)
172.16.0.0/24 subnet
public agent VM scale set
role: slave_public
http:80
https:443
8080
Docker:2375
agent0
(10.0.0.4)
10.0.0.0/8 subnet
private agent VM scale set
Docker:2375
agent0
(10.32.0.4)
10.32.0.0/8 subnet

kubernetes


22
443
NAT – public IP
Loadbalancer
Azure LoadBalancers (dynamically added to service)
SSH
443
private vnet 10.0.0.0/8
Master-0
host addr: 10.240.255.5
pod CIDR: 10.244.0.0/24
443
scheduler
apiserver
etcd
controller-manager
Proxy
Implements Service network: 10.0.0.0/16
kubelet
Docker
availability set
Node-0
host addr: 10.240.0.4
pod CIDR: 10.244.1.0/24
Proxy
Implements Service network: 10.0.0.0/16
kubelet
Docker
Node-1
host addr: 10.240.0.5
pod CIDR: 10.244.2.0/24
Proxy
Implements Service network: 10.0.0.0/16
kubelet
Docker

Single command:

```
az acs create
 -n "acs-cluster"
 -g “ClusterResourceGroup“
 -d “myclusterdns"
 --orchestrator-type "Kubernetes"
 --master-count 1
 --agent-count 3
 --generate-ssh-keys
```

Azure Container Service Engine (acs-engine)

- Generates ARM (Azure Resource Manager) templates
  - Choice of DC/OS, Kubernetes, or Swarm orchestrators
  - Multiple agent pools where each agent pool can specify:
    - VM sizes
    - Node count
    - Virtual Machine Scale Sets or Availability Sets
- Clusters with 100+ nodes
- Custom virtual network configuration

https://github.com/Azure/acs-engine/

DEMO

Deploying Azure Container Service

Visual Studio Team Services

Visual Studio
Team Services

iOS

Team Foundation
Server

iOS

Plan & Track Work

Source Code Management

Package Management

Quality Management

Cross-platform Build

Continuous Deployment

Release Management

Feedback Management

Application Telemetry

Extend and Customize

http://www.visualstudio.com/news/release-archive-vso

GitHub
JetBRAINS Develop with pleasure!
LeanKit
AppVeyor
slack
Octopus Deploy
azuqua
myget
Jenkins
HU-BOT
Atlassian HipChat
PreEmptive Solutions
Tasktop
SOASTA
CloudBees
dynatrace
zendesk
MODERN Requirements By eDevTECH

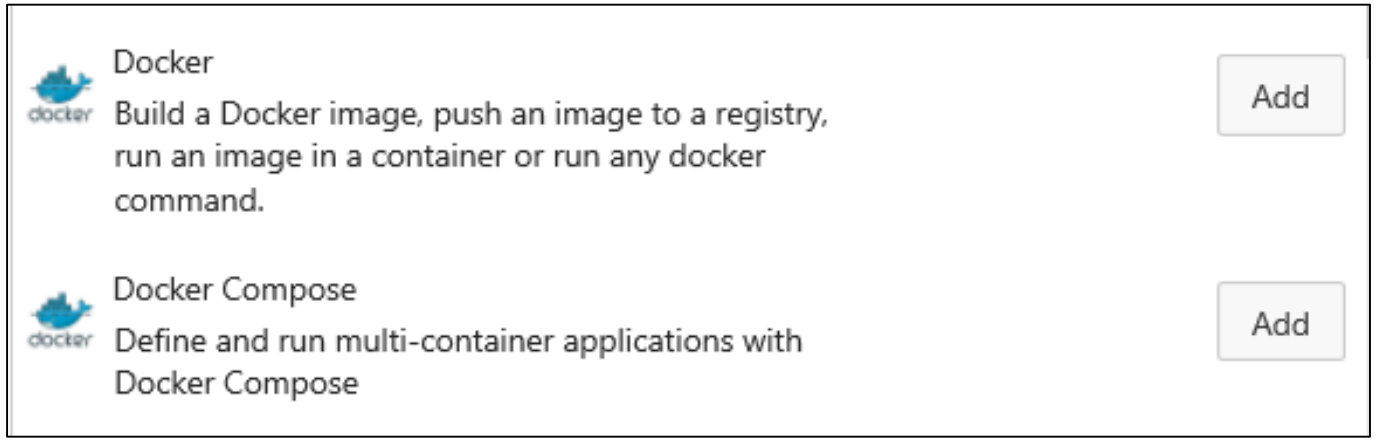
Open + Flexible Tool Chain

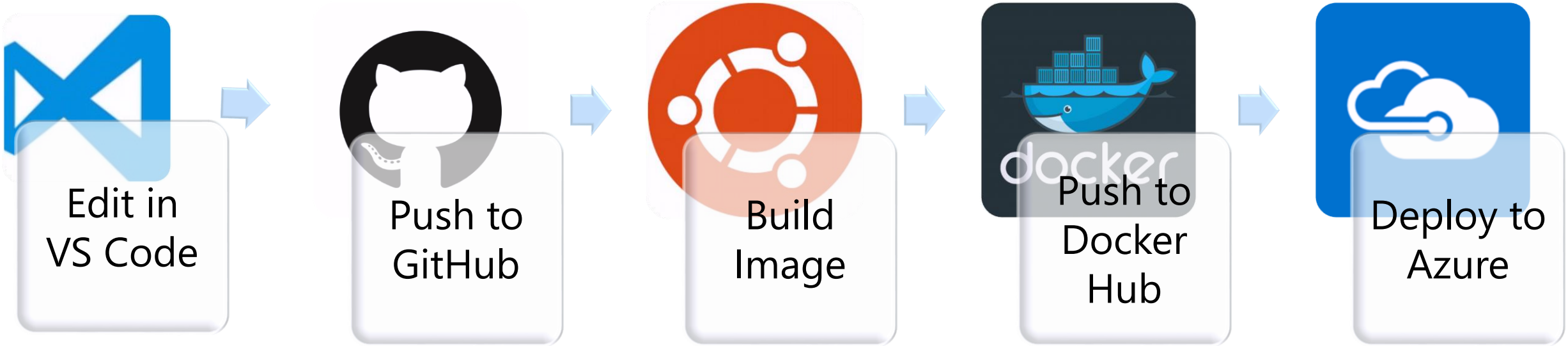
1,177	6,538	90	171
Visual Studio Code Extensions	Visual Studio Gallery Extensions	Visual Studio Sim-Ship Partners	VS Team Services Extensions

Aha!
Trello
IncrediBuild BEYOND ACCELERATION
zapier
macincloud
uservoice
flowdock
bamboo solutions
VS Anywhere
Xamarin
jamo solutions
perfecto mobile
TFS TIMETRACKER
redgate ingeniously simple
CHEF
Campfire
submain
EKOBIT
sauce LABS
OpsHub Enabling Agile Integration

VSTS and Docker Demo

Using the Visual Studio Team Services and the Docker extensions.

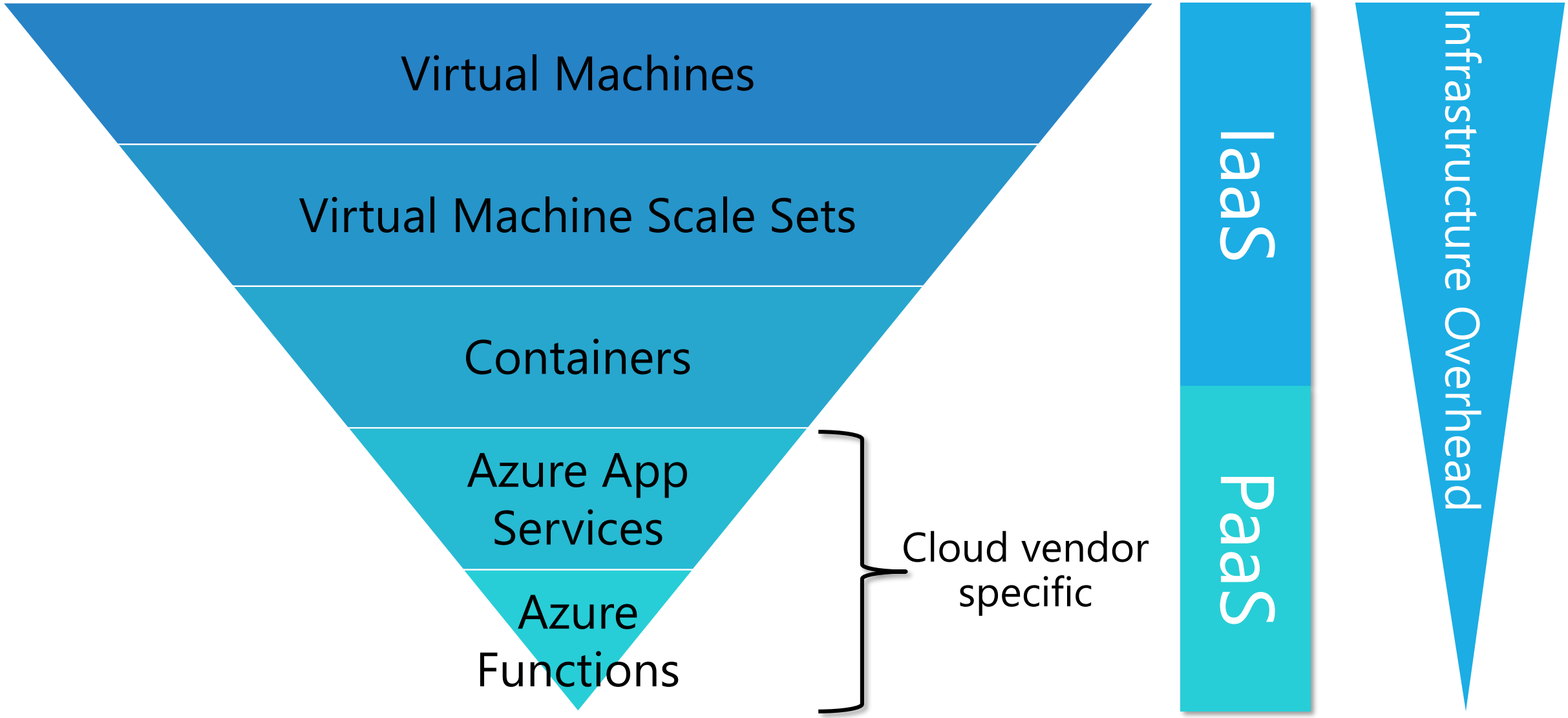




DEMO

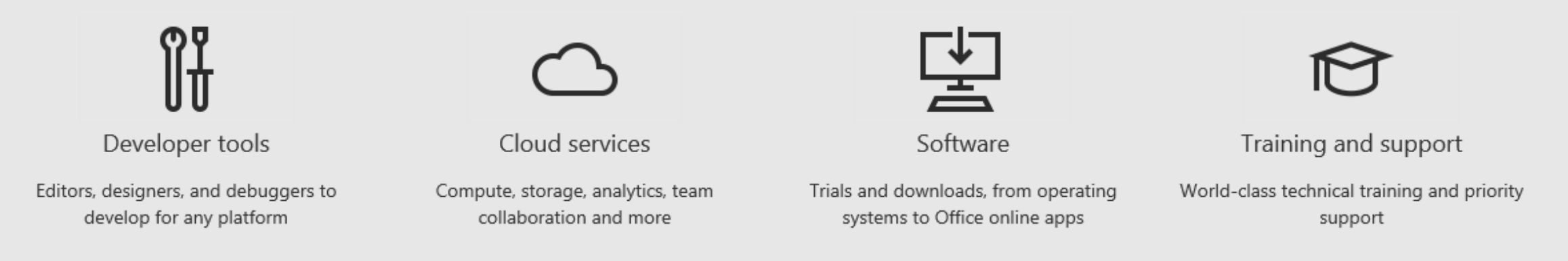
VSTS and Docker

Containers are an option…



Want to try this out?

Visual Studio Dev Essentials: https://developer.microsoft.com



IT Pro Cloud Essentials: https://www.itprocloudessentials.com

Technical resources:

- DevOps Fundamentals: https://channel9.msdn.com/Series/DevOps-Fundamentals
- DevOps Dimensions: https://channel9.msdn.com/Shows/DevOps-Dimension
- The Container Channel https://channel9.msdn.com/Blogs/containers
- Get access to free online training: https://mva.microsoft.com/training-topics/devops
- Slides and demos: https://github.com/marrobi/Microsoft-and-Containers

marrobi@microsoft.com
@techdiction

Questions?

© 2016 Microsoft Corporation. All rights reserved.