



Microsoft & Containers

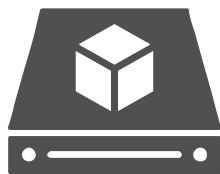
Marcus Robinson
Technical Evangelist
@techdiction



Containers Overview

Containers

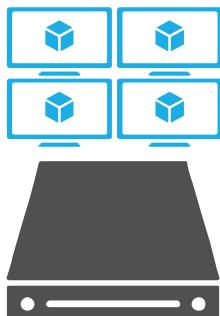
A new approach to build, ship, deploy, and instantiate applications



Physical

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

New applications often required new physical systems for isolation of resources



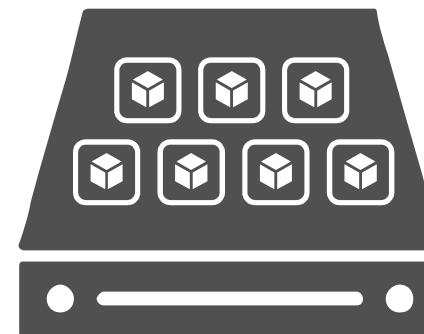
Virtual

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



Physical/virtual

Package and run apps within containers

Key benefits

Further accelerate app deployment

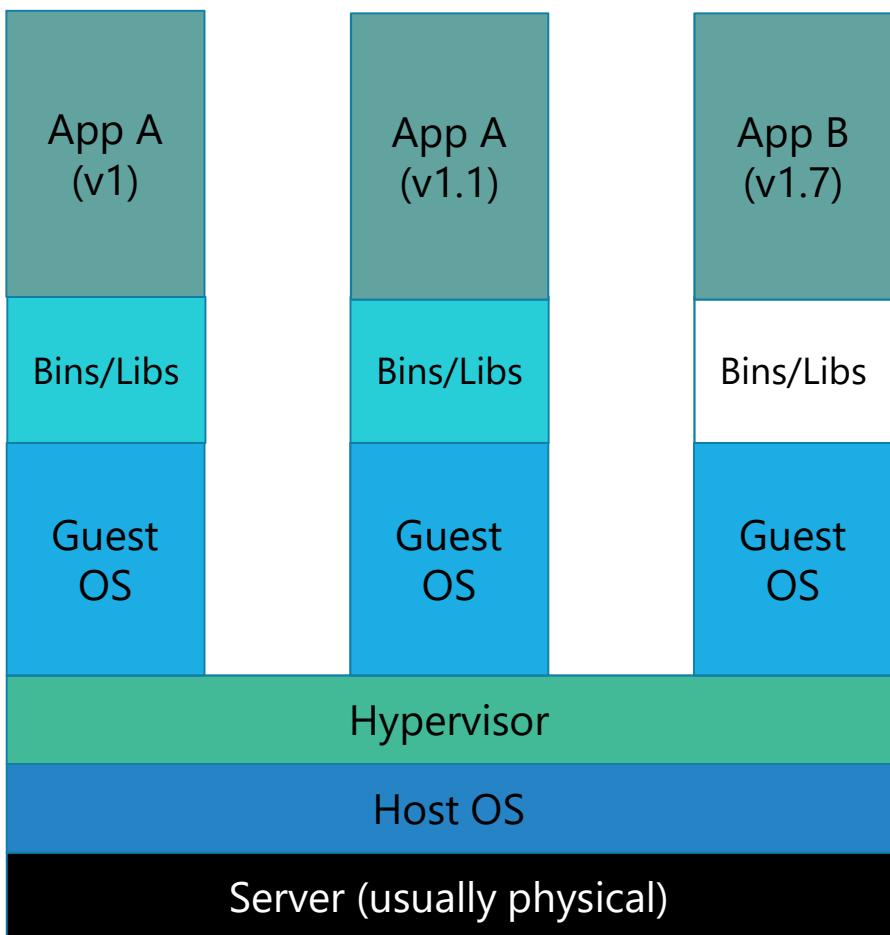
Streamline development and testing

Reduces the "it works on my machine" problem

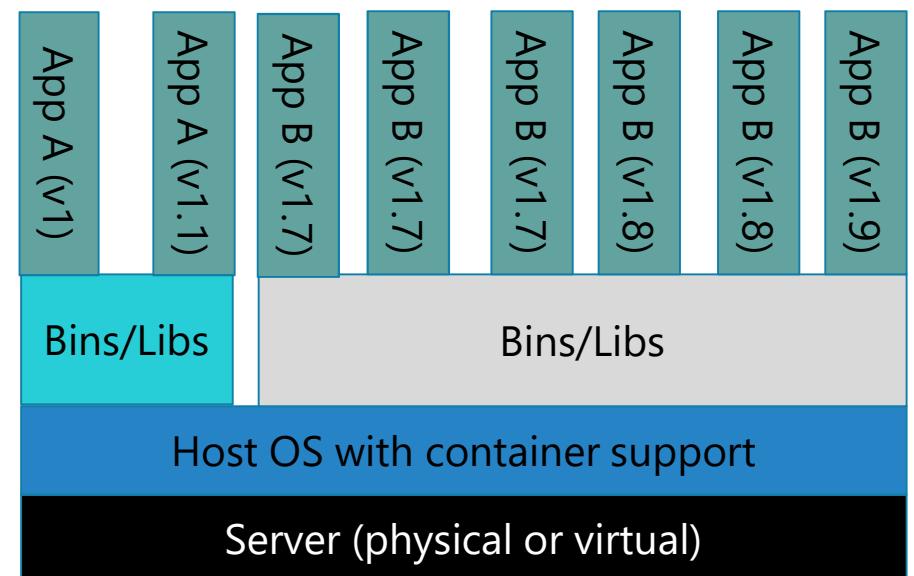
Lower costs associated with app deployment

Increase server consolidation

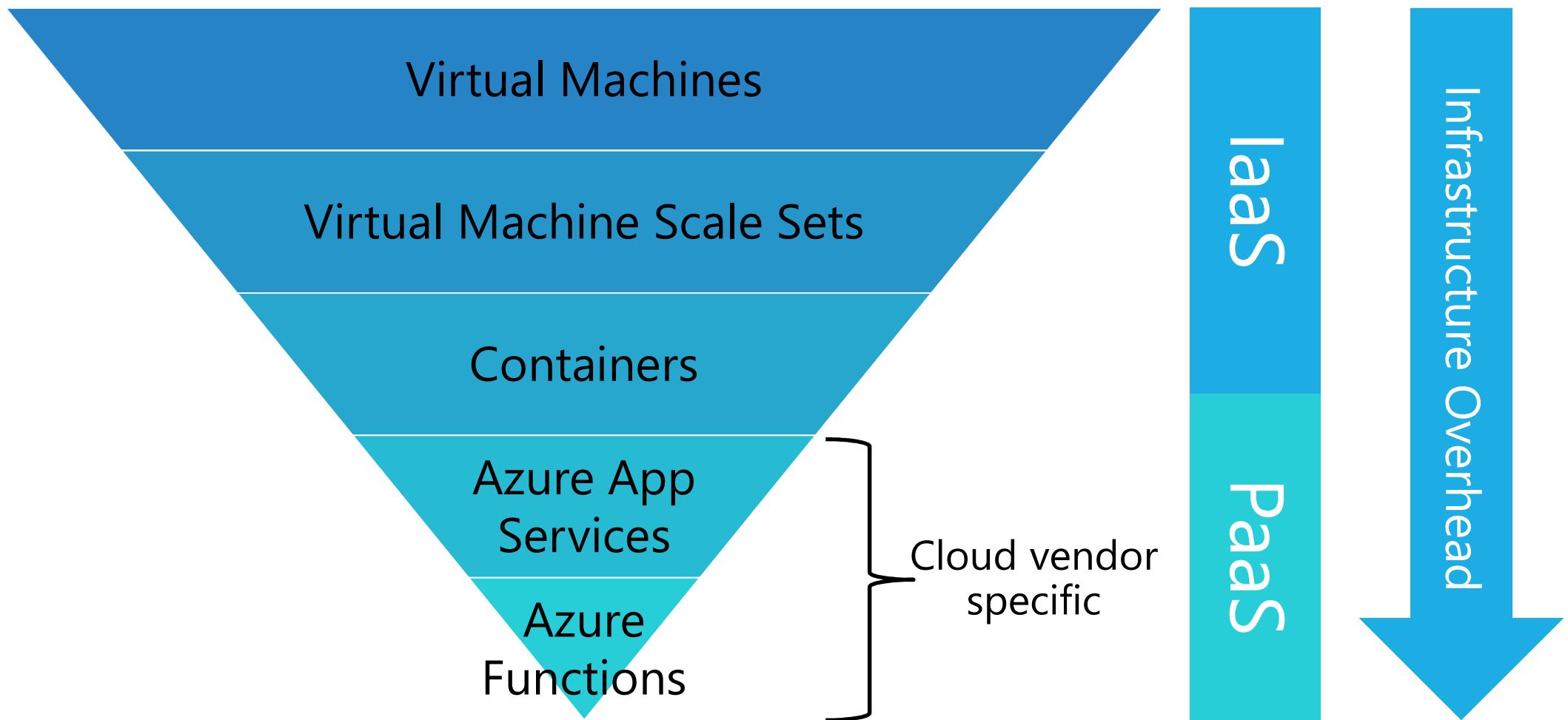
Server Virtualisation: Each app and each version of an app has dedicated OS



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



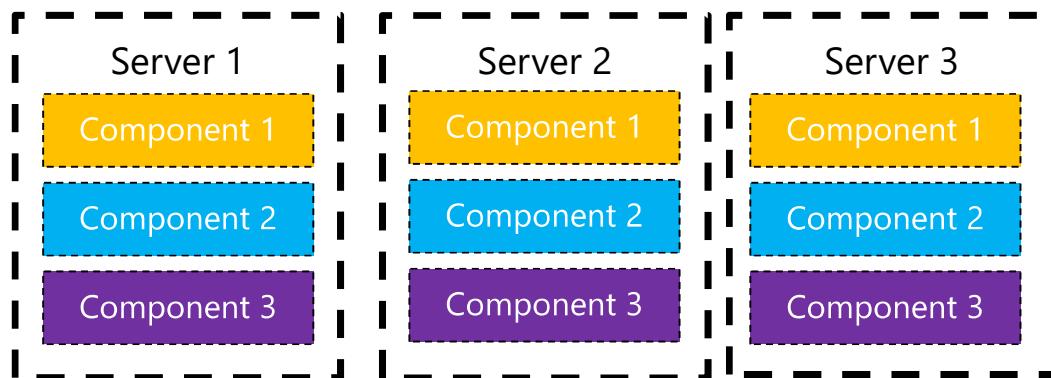
Where do containers fit?



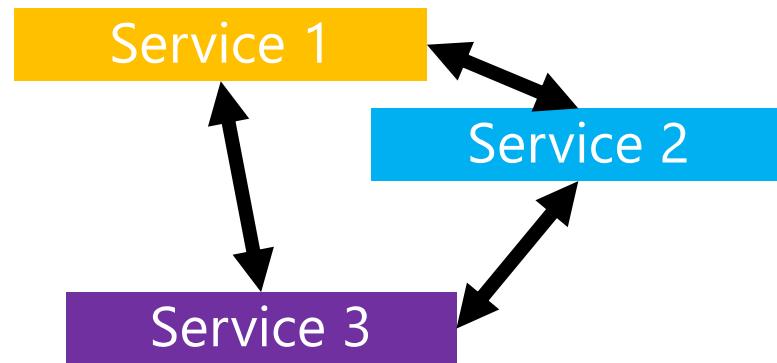
Monolithic Applications



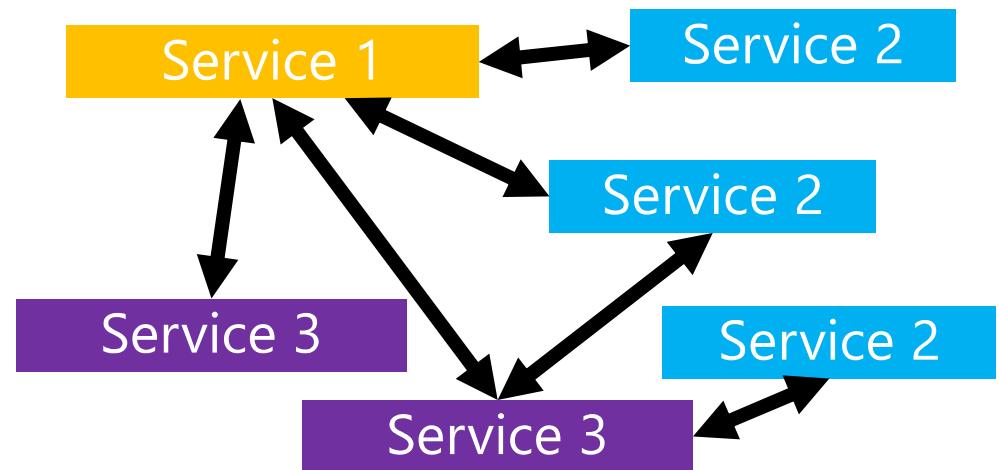
Scaled Monolith



Microservices

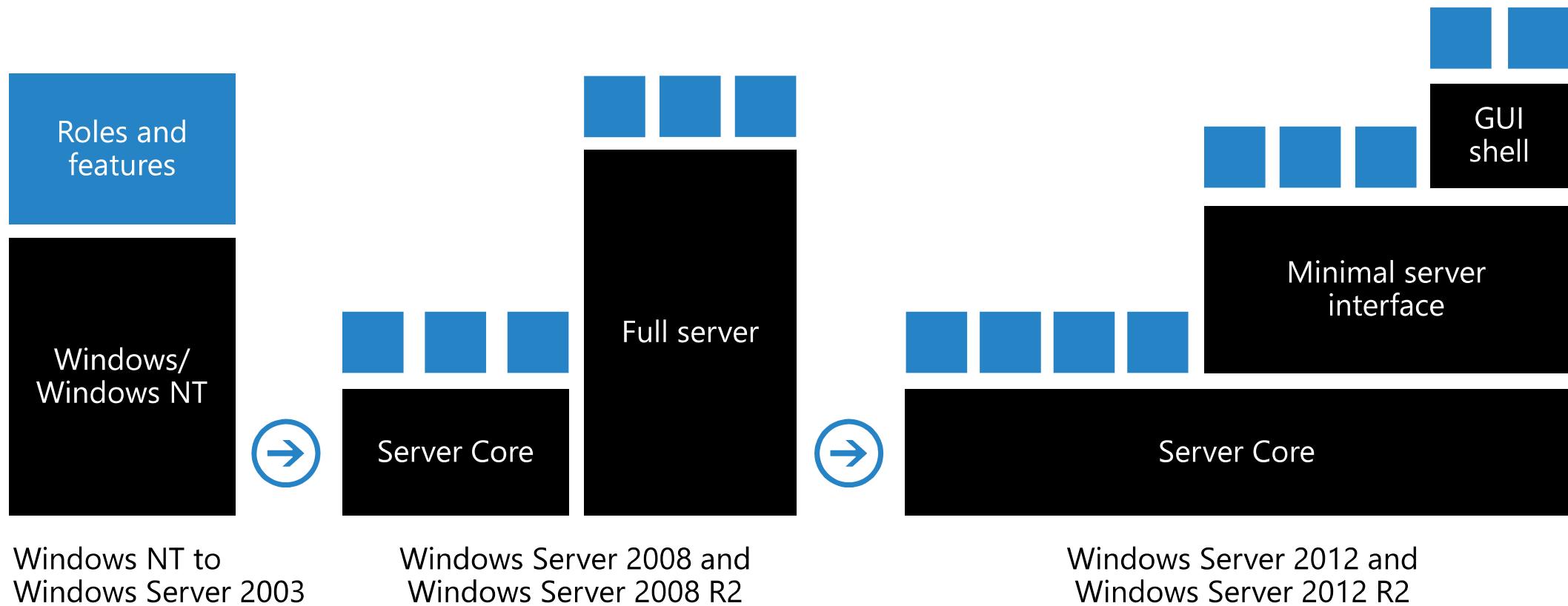


Scaled Microservices



Windows Server Containers

The story so far...



The next step in the journey...

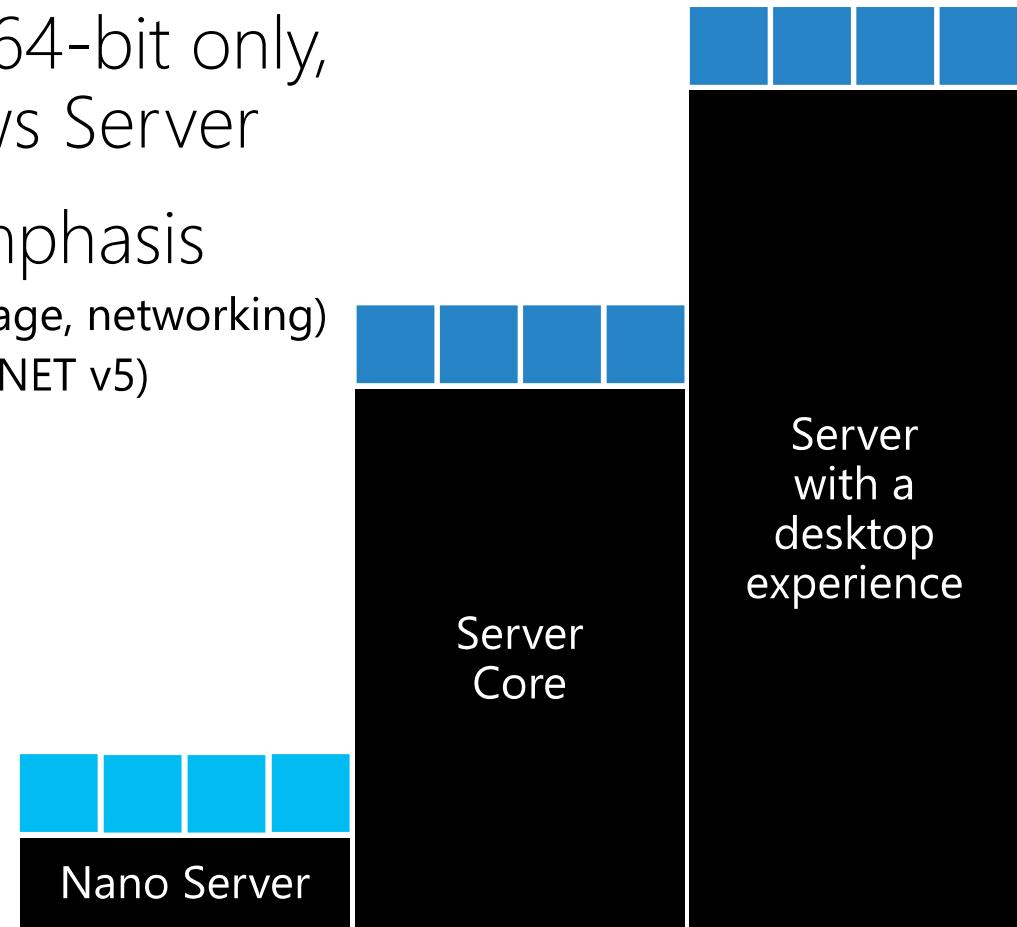
→ Nano Server: A new headless, 64-bit only, deployment option for Windows Server

→ Deep refactoring with cloud emphasis

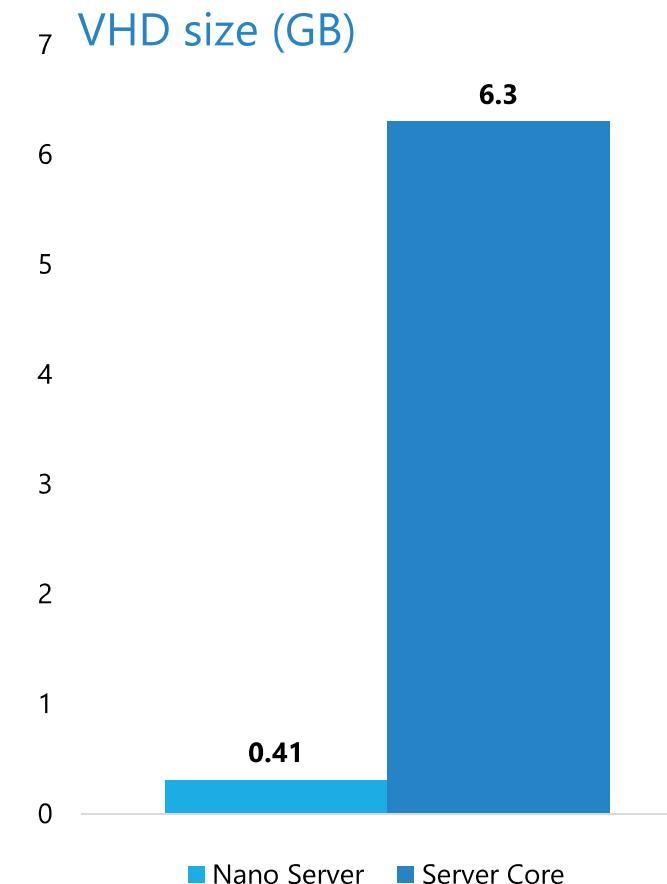
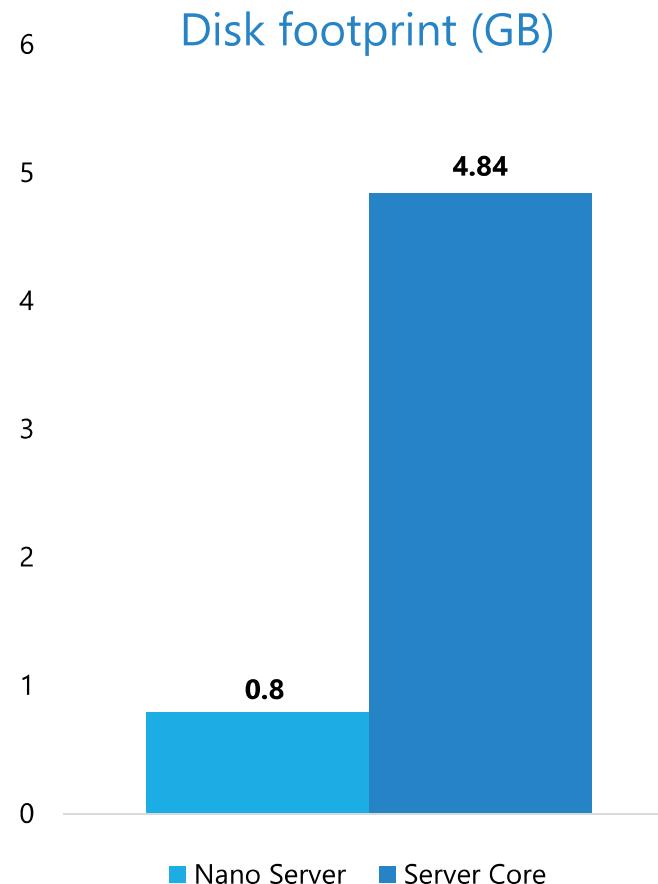
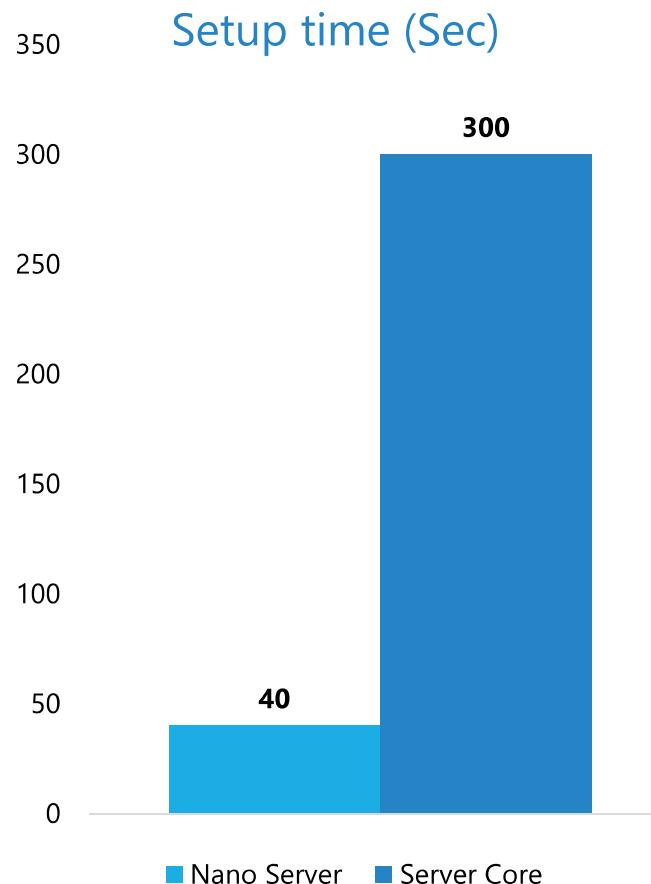
- Cloud fabric & infrastructure (clustering, storage, networking)
- Born-in-the-cloud applications (PaaS v2, ASP.NET v5)
- VMs & Containers (Hyper-V & Docker)

→ 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



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

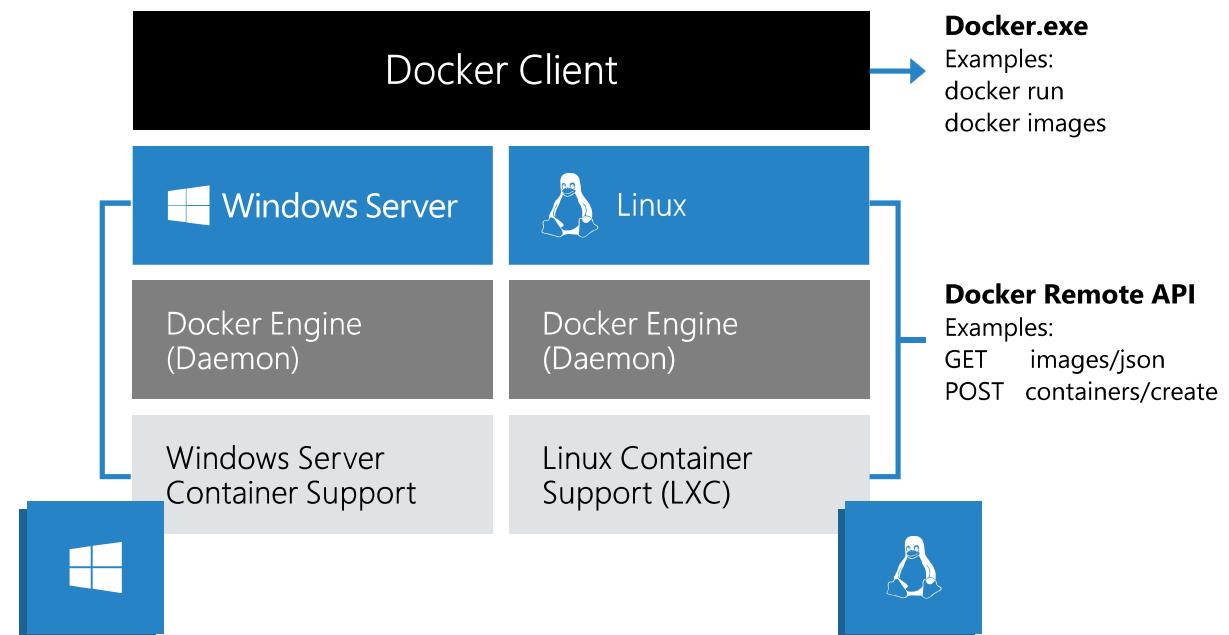
Bring Windows Server containers to the Docker ecosystem to expand the reach of both developer communities

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

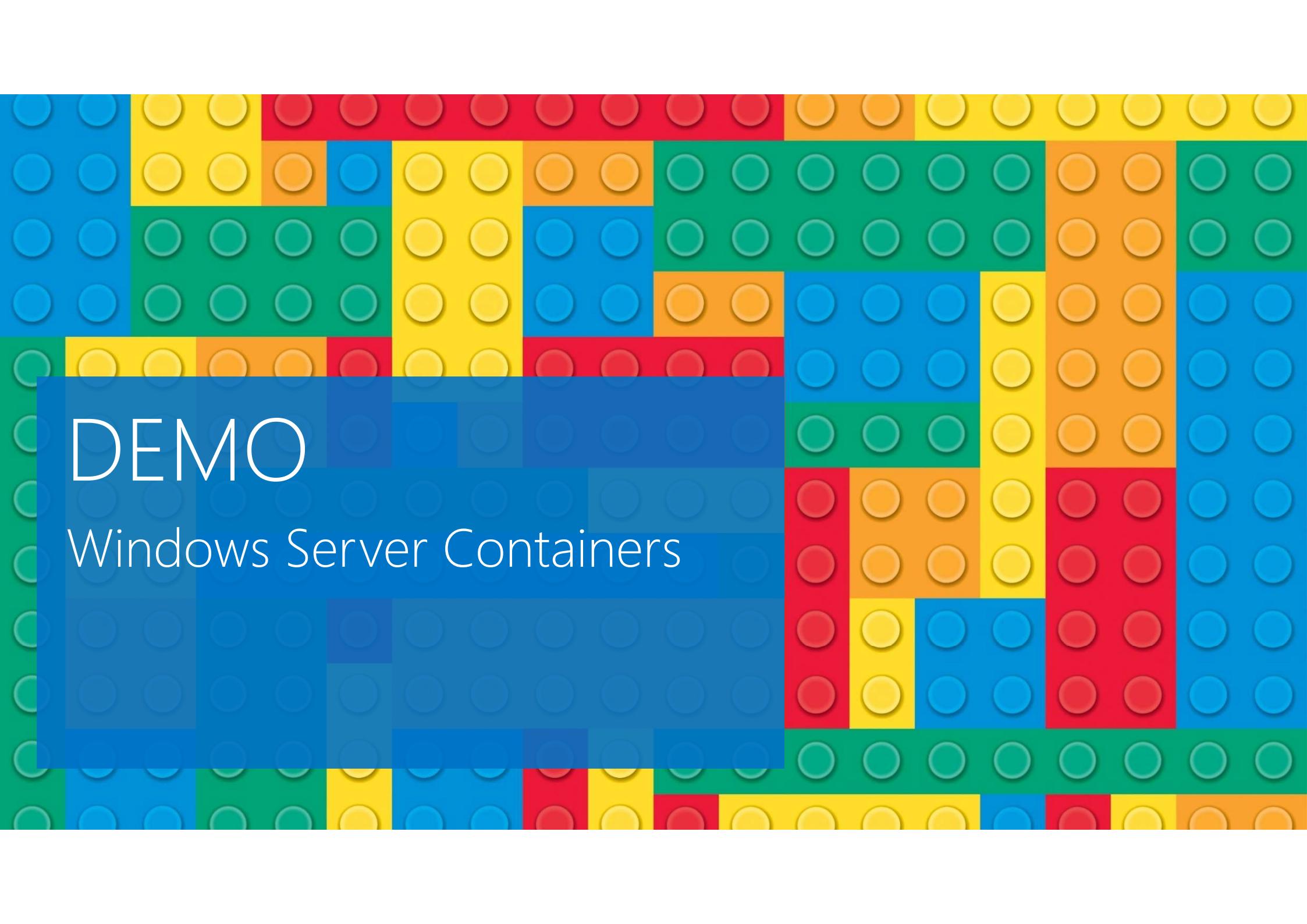


PowerShell

- Containers can also be managed using PowerShell
- A open source project:
<https://github.com/Microsoft/Docker-PowerShell/>
- Executes commands against the Docker Engine API as specified by the (DOCKER_HOST) environment variable

Example:

Get-Container | Stop-Container



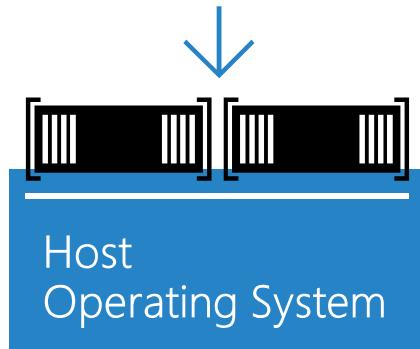
DEMO

Windows Server Containers

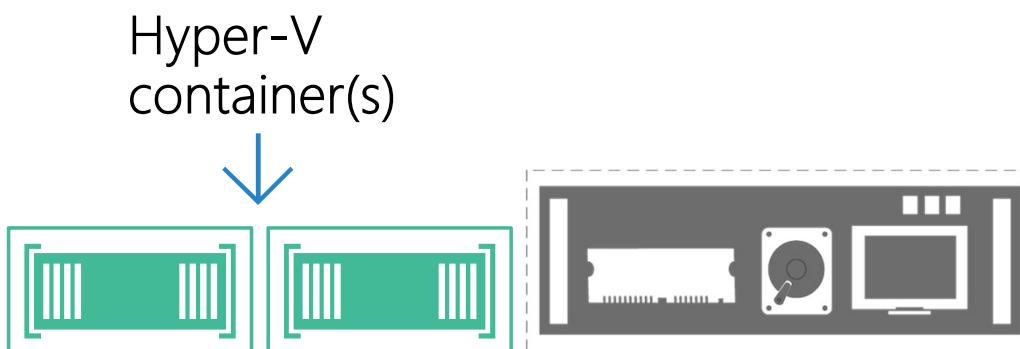
Hyper-V Containers

Hyper-V Containers

Windows Server
container(s)



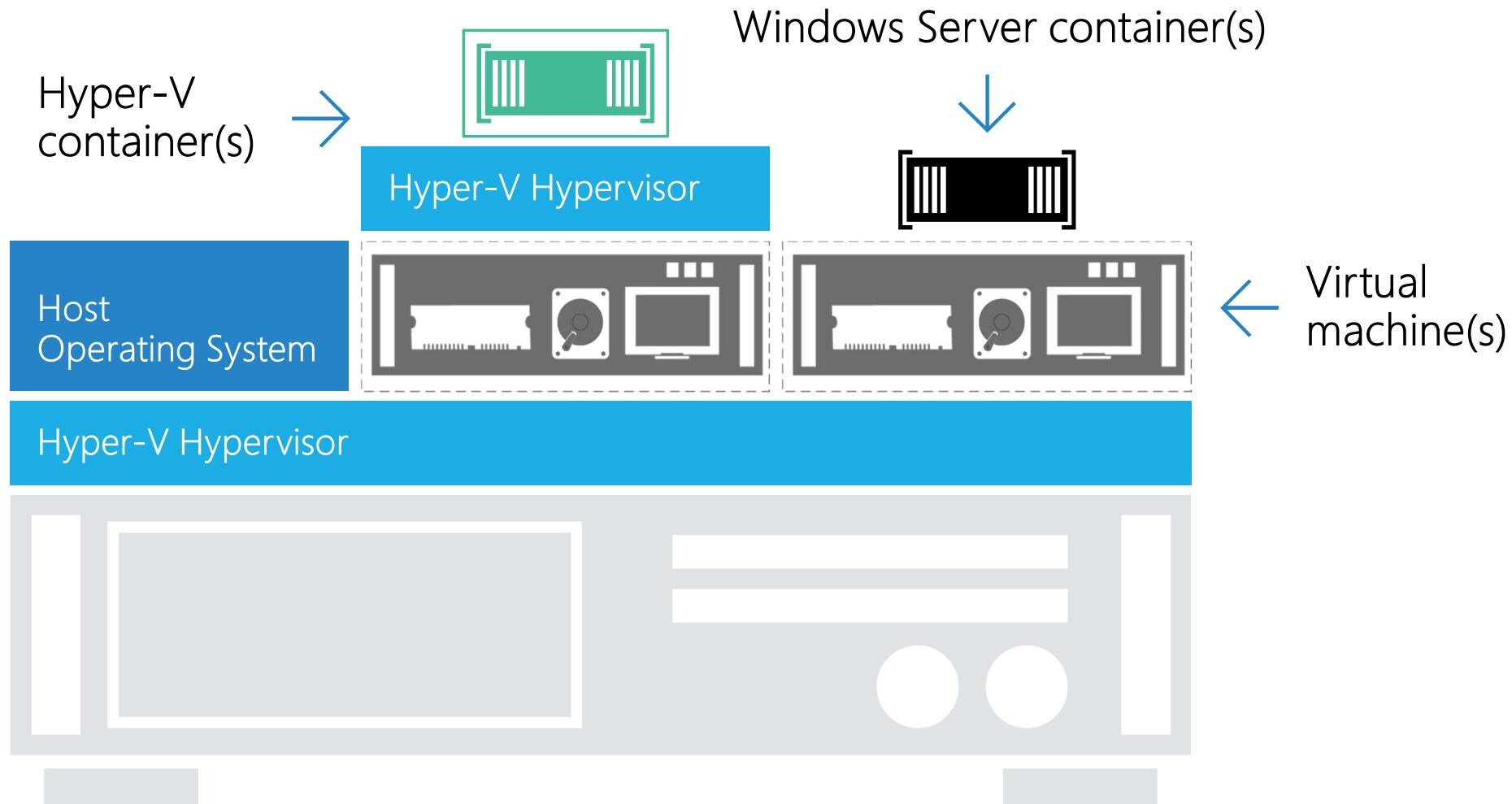
Hyper-V
container(s)



Virtual
machine(s)



Hyper-V Containers using nested virtualization



Creating Hyper-V Containers:

`Install-NanoServerPackage Microsoft-NanoServer-Compute-Package`

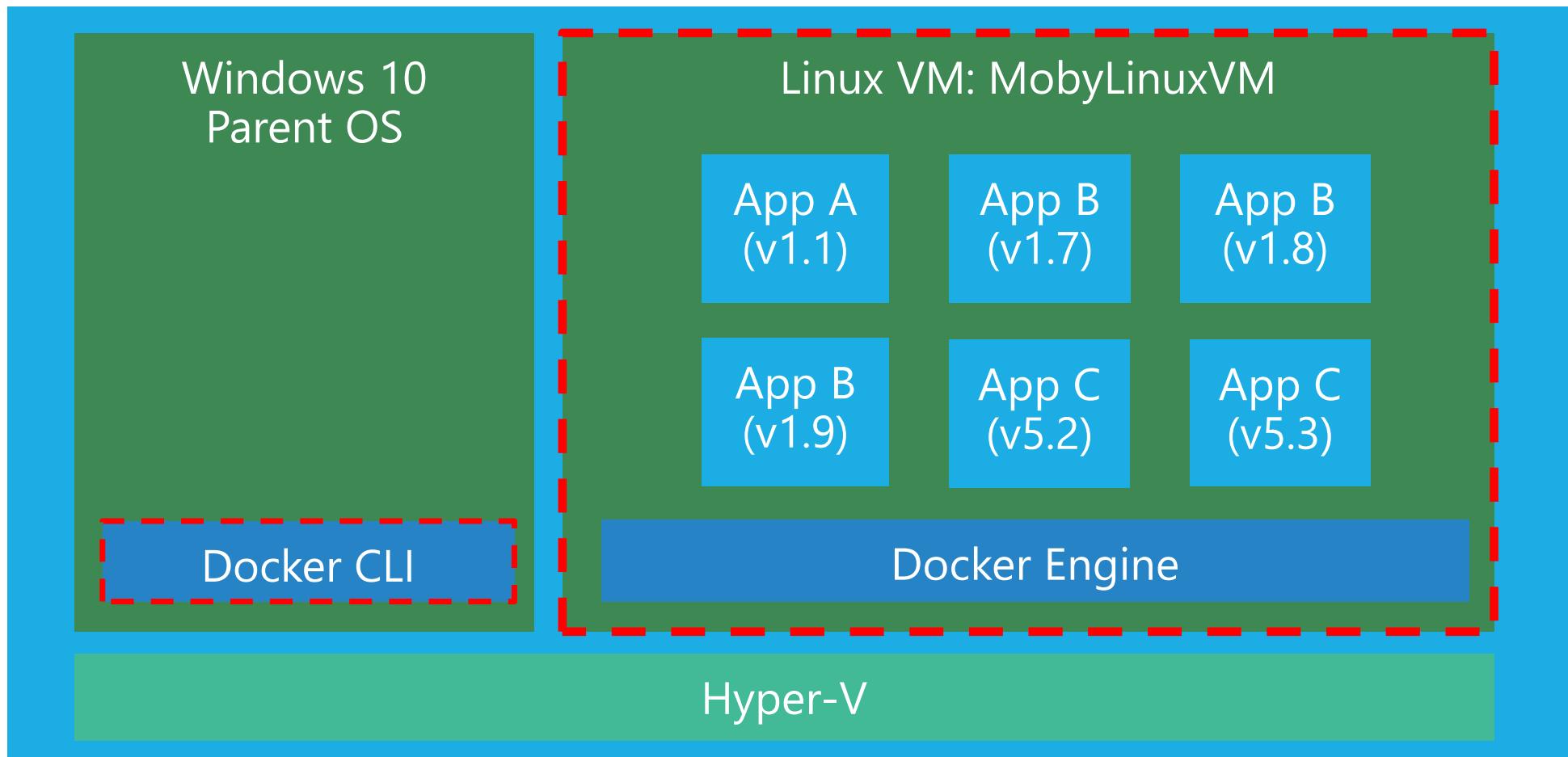
```
docker run -d --isolation=hyperv nanoserver cmd
```

Host Operating System	Windows Server container	Hyper-V container
Windows Server 2016 Full UI	Server Core image	Nano Server image
Windows Server 2016 Core	Server Core image	Nano Server image
Windows Server 2016 Nano	Nano Server image	Nano Server image
Windows 10	Not Available	Nano Server image

Hyper-V is the default and only option on Windows 10

Docker for Windows

Docker for Windows



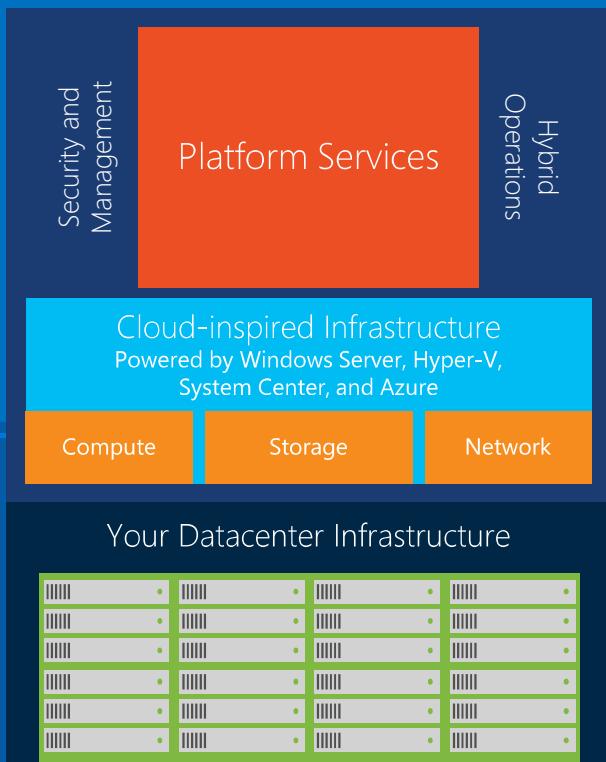


DEMO

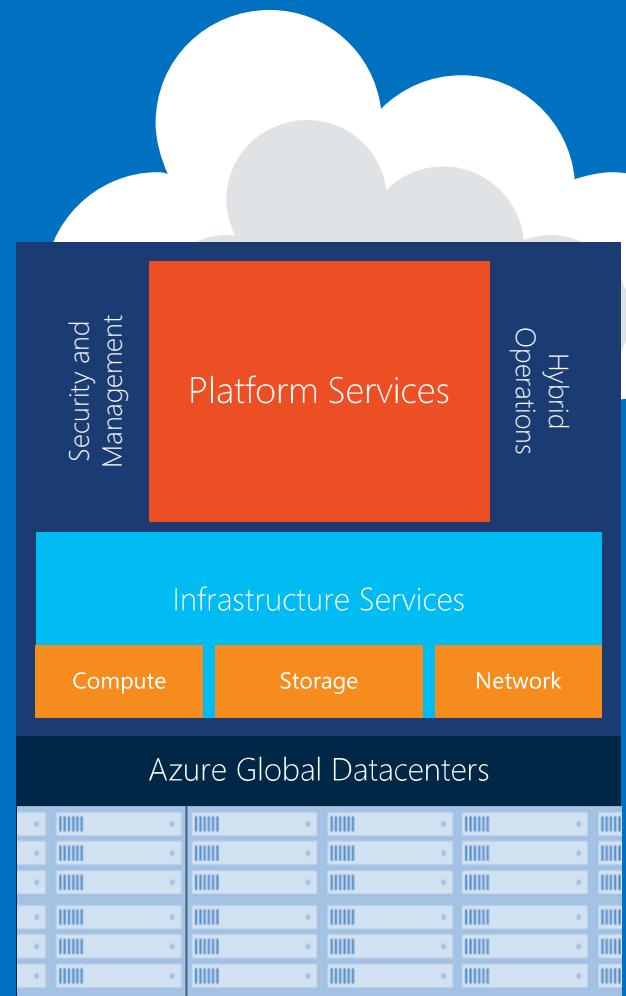
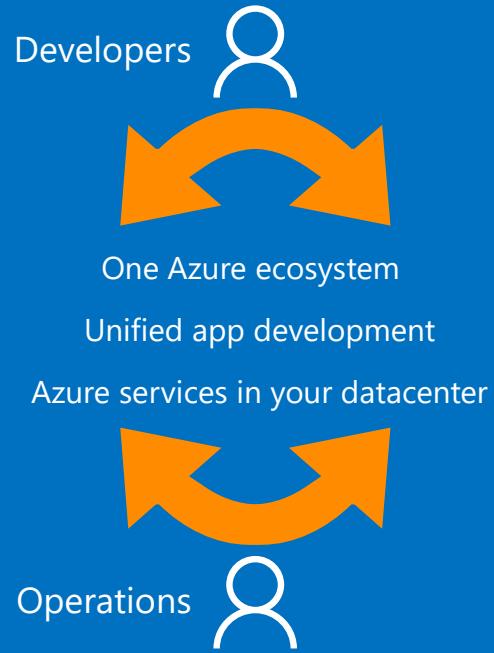
Docker for Windows

Azure and Containers

Microsoft Azure & Azure Stack



Microsoft Azure Stack
and Cloud Platform System



Microsoft Azure Cloud
Public, Global, Shared

Open cloud

Hyper-scale

Proven

Hybrid

Open & flexible

Management



Applications



App Frameworks



Databases & Middleware



Infrastructure



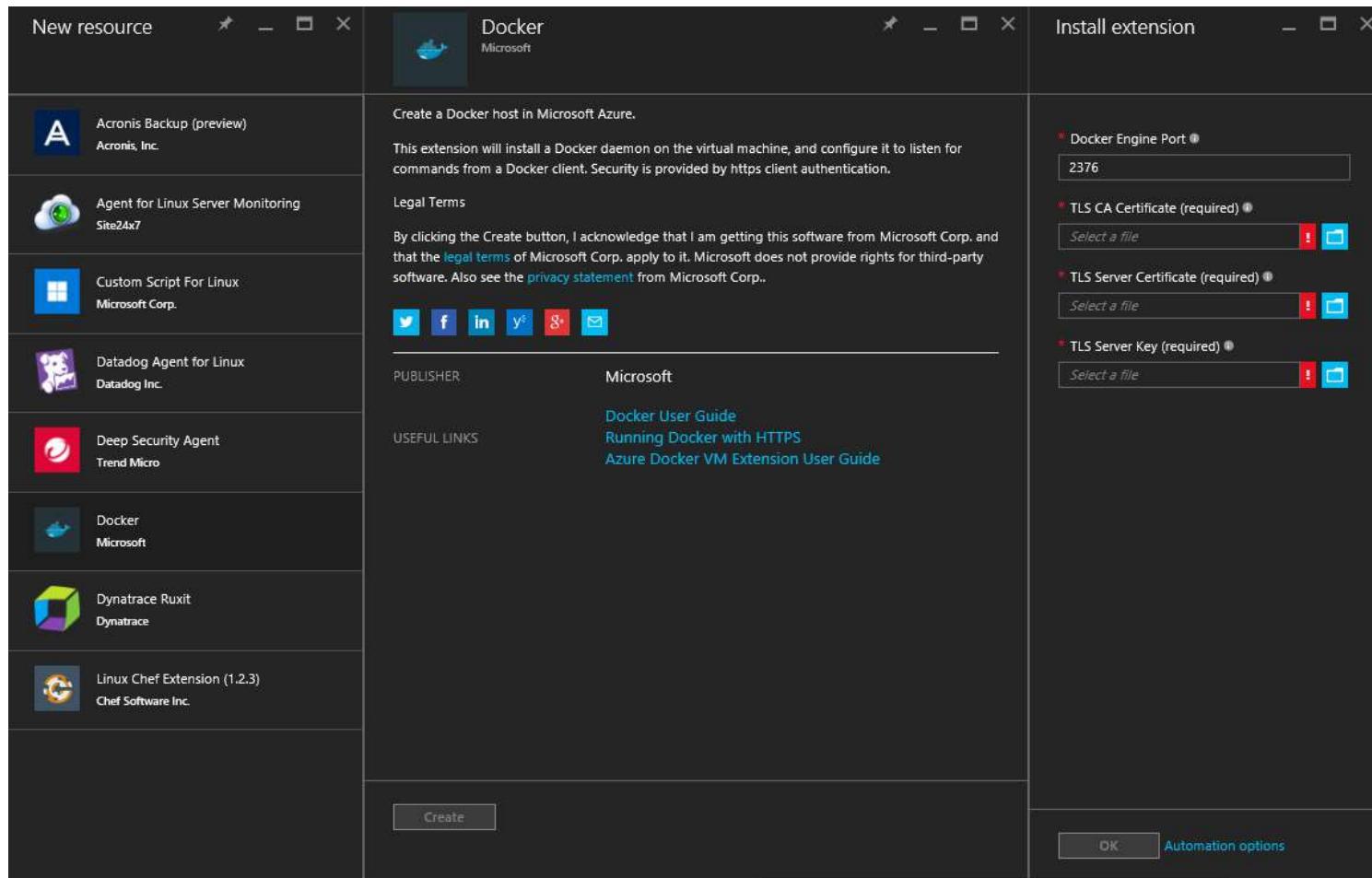
Marketplace VMs using Docker

docker		
 Docker on Ubuntu Server	Canonical + Microsoft	Virtual Machines
 ghost	Docker	Virtual Machines
 crate	Docker	Virtual Machines
 rabbitmq	Docker	Virtual Machines
 elasticsearch	Docker	Virtual Machines
 wordpress + mysql	Docker	Virtual Machines
 Redis-datastore	Docker	Virtual Machines
 mysql	Docker	Virtual Machines

Docker VM Extension

<https://github.com/Azure/azure-docker-extension>

- Install Docker on a Linux VM using Portal, PowerShell or ARM Template



Docker VM Extension in ARM Template

```
{  
  "type": "Microsoft.Compute/virtualMachines/extensions",  
  "name": "[concat(variables('vmName'), '/', variables('extensionName'))]",  
  "apiVersion": "2015-05-01-preview",  
  "location": "[resourceGroup().location]",  
  "dependsOn": [  
    "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"  
  ],  
  "properties": {  
    "publisher": "Microsoft.Azure.Extensions",  
    "type": "DockerExtension",  
    "typeHandlerVersion": "1.0",  
    "autoUpgradeMinorVersion": true,  
    "settings": {}  
  }  
}
```

Deploy services by providing
docker-compose file in JSON format

```
--  
  "protectedSettings": {  
    "environment": {  
      "MYSQL_ROOT_PASSWORD": "[parameters('mysqlPassword')]"  
    }  
  },  
  "settings": {  
    "compose": {  
      "db": {  
        "image": "mysql",  
        "ports": [  
          "3306:3306"  
        ],  
        "volumes": [  
          "/var/lib/mysql:/var/lib/mysql"  
        ],  
        "environment": [  
          "MYSQL_ROOT_PASSWORD"  
        ]  
      },  
      "wordpress": {  
        "image": "wordpress",  
        "ports": [  
          "80:80"  
        ],  
        "links": [  
          "db:mysql"  
        ]  
      }  
    }  
  }  
}
```

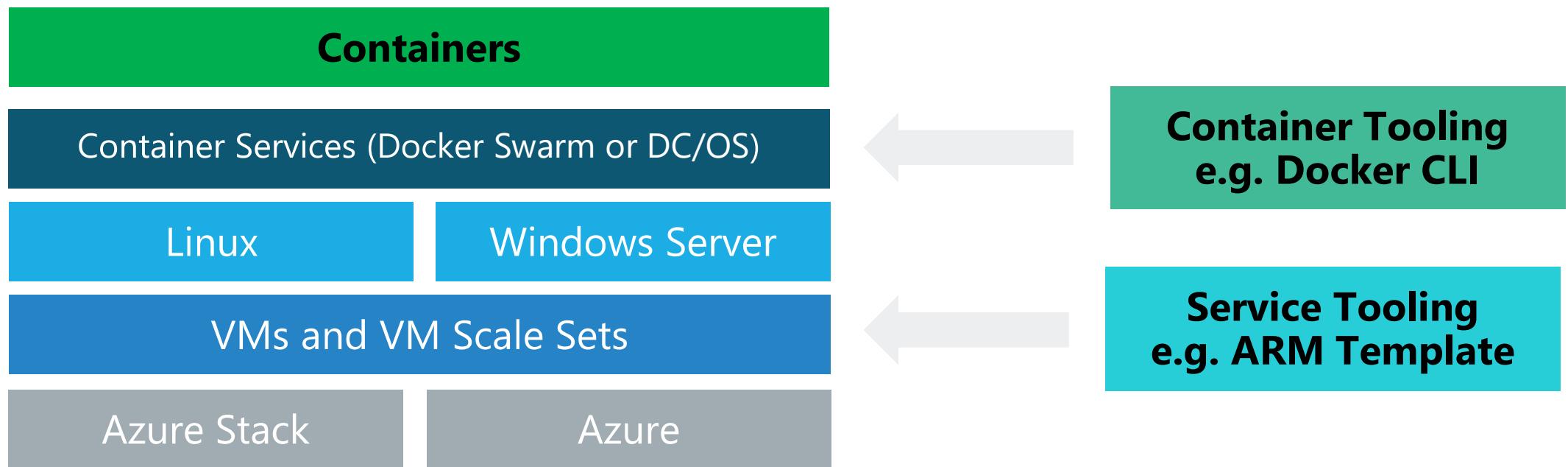
Azure Driver for Docker Machine

- Create a Linux VM with Docker using a single command

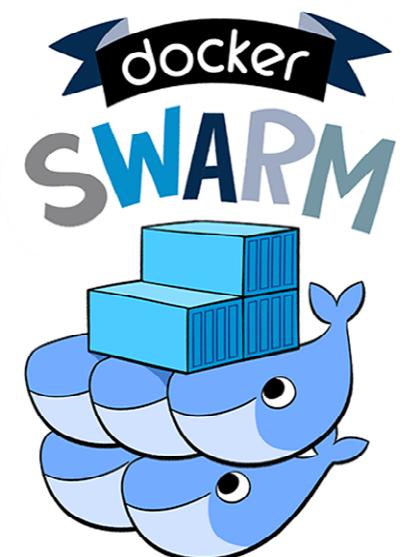
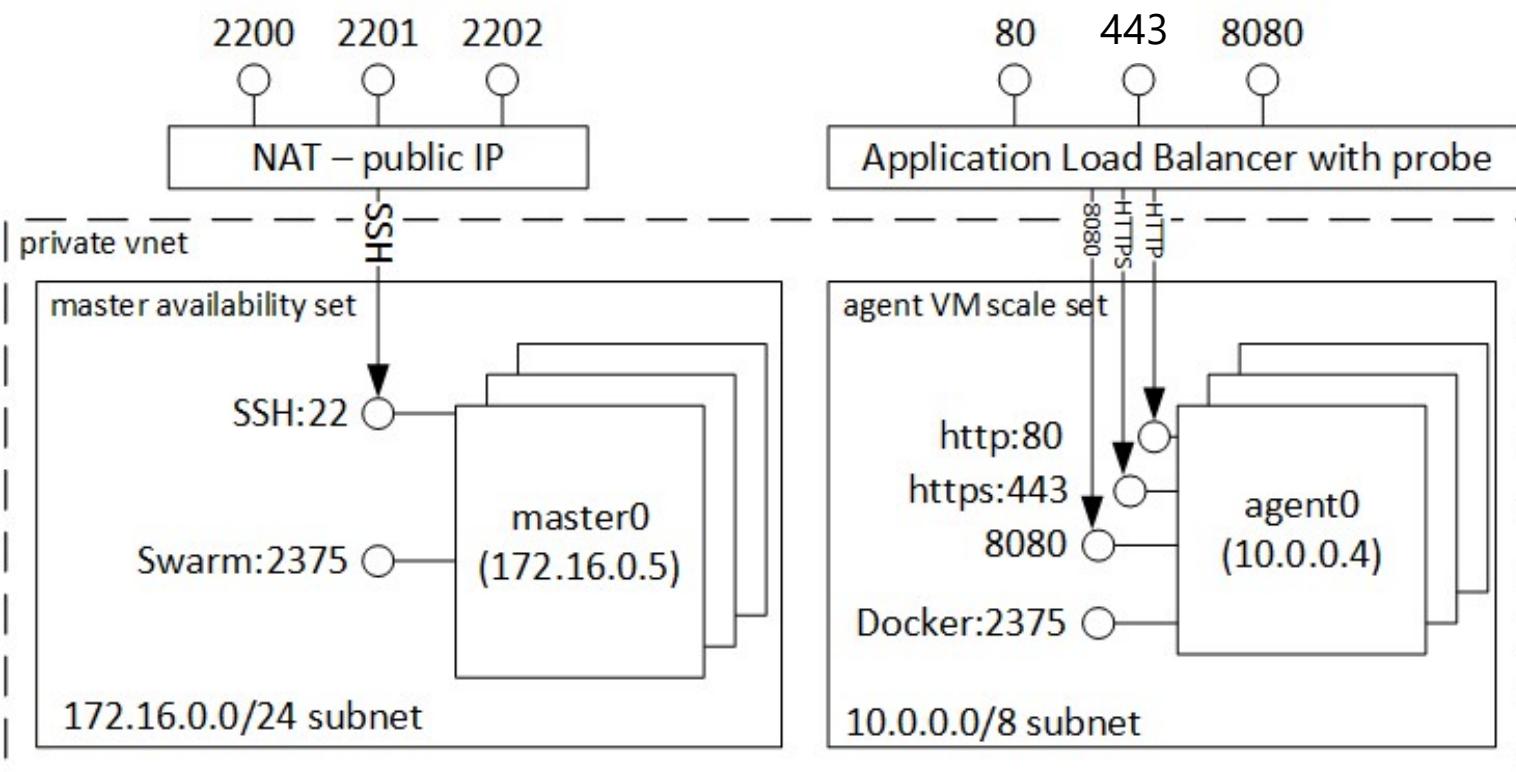
```
$DockerMachineVMName = "dockerdemovm"  
  
docker-machine create --driver azure  
  --azure-subscription-id $SubscriptionID  
  --azure-location 'West Europe'  
  --azure-resource-group 'DevOpsDemo-DockerMachine'  
  $DockerMachineVMName
```

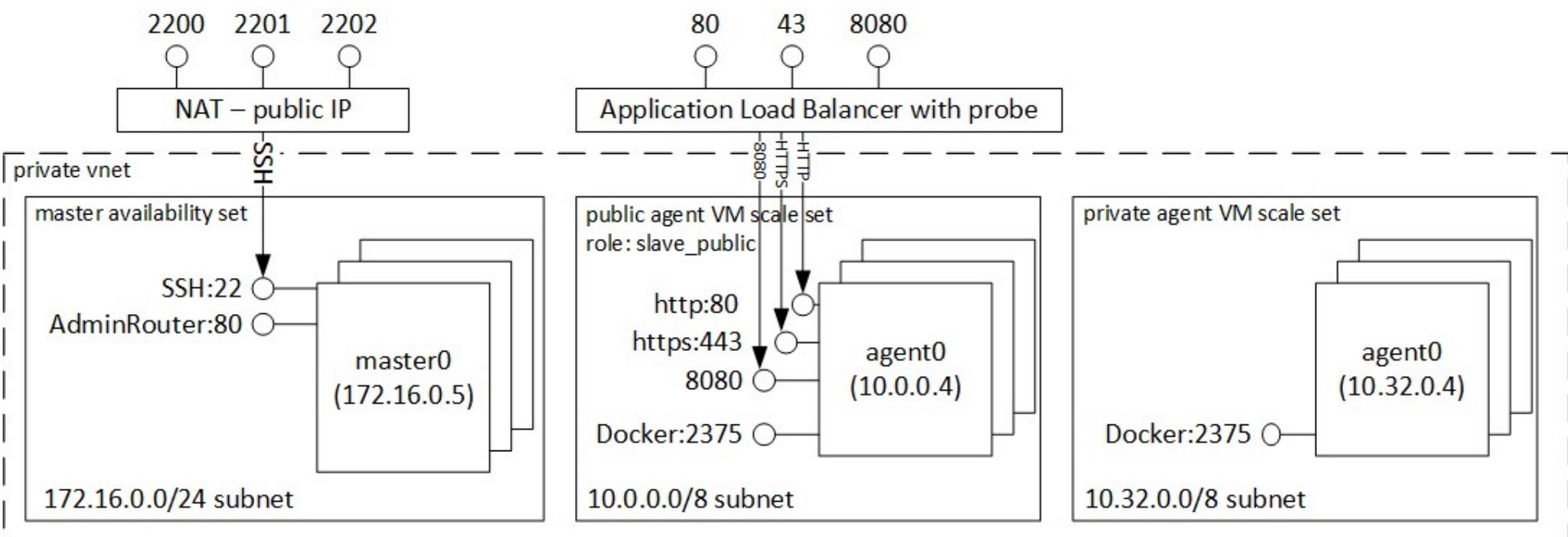
<https://docs.docker.com/machine/drivers/azure/>

Azure Container Service

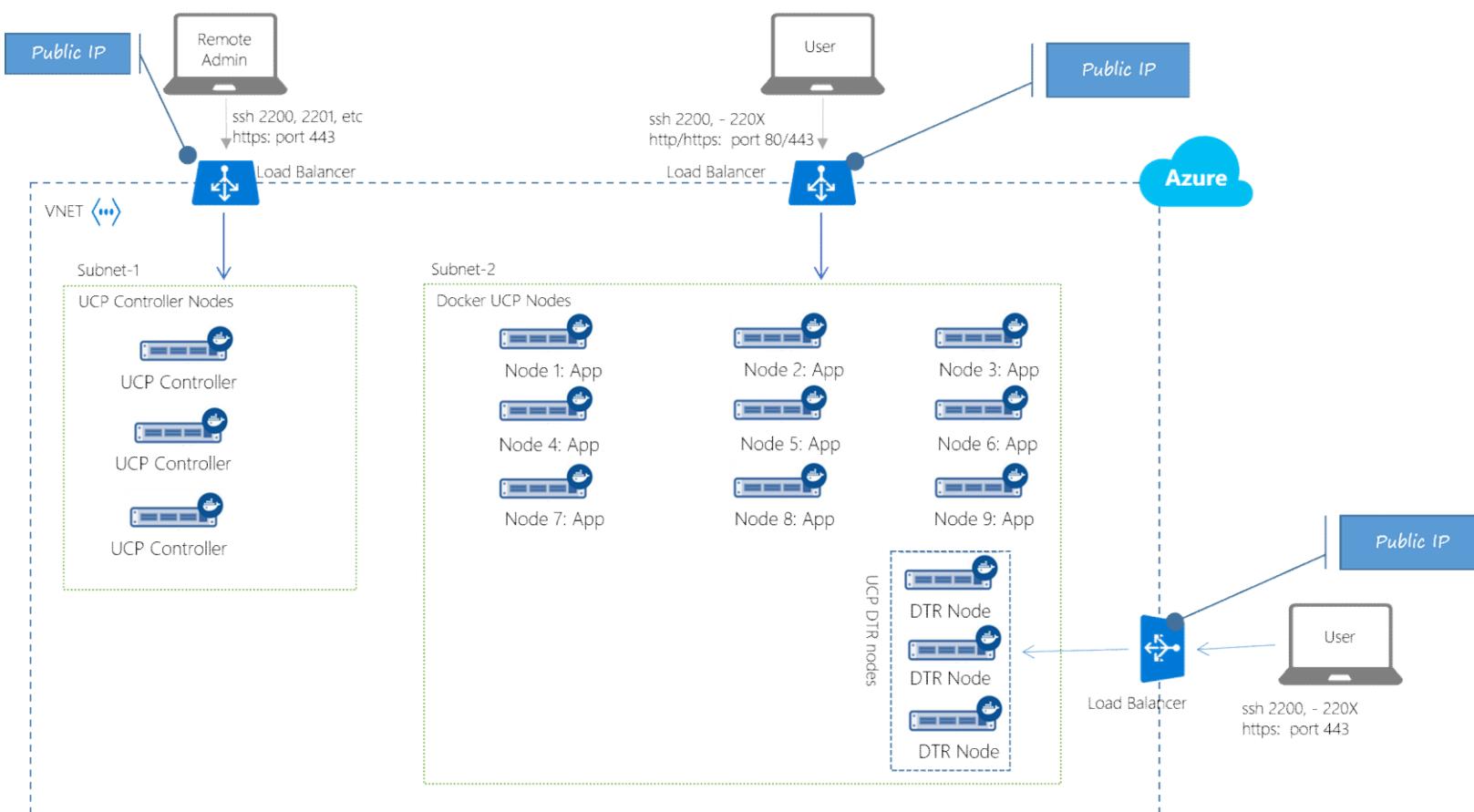


ACS with Swarm



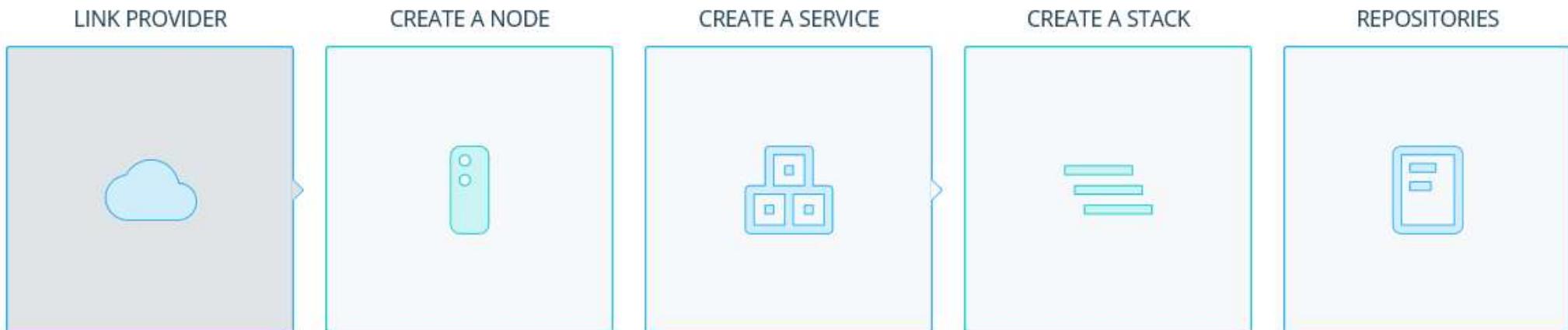
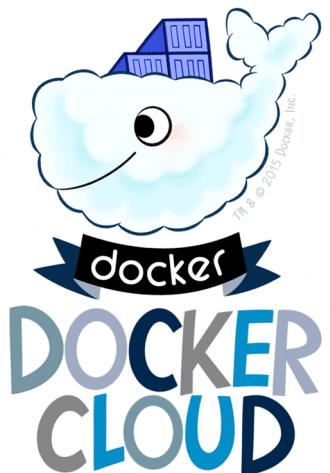


Docker Datacentre



Docker Cloud

- Docker hosted solution
- Nodes can run anywhere
- Pay per node



Link to a hosted Cloud Services Provider like DigitalOcean or AWS.

*A **node** is a Linux host or virtual machine used to deploy and run containers.*

*A **service** is a container, or a group of containers from the same Docker repository. You can add more containers to scale an app across nodes.*

*A **stack** specifies a group of services that make up an application, similar to Docker Compose.*

*A **repository** is a collection of tagged images. When you create a service, you choose an image to use to create containers.*

Docker for Azure

- Existing Docker implementations require rules adding to Azure load balancer to publish services
- Docker for Azure automatically adds the appropriate rules when a container or service is published
- Sign up at <http://beta.docker.com>

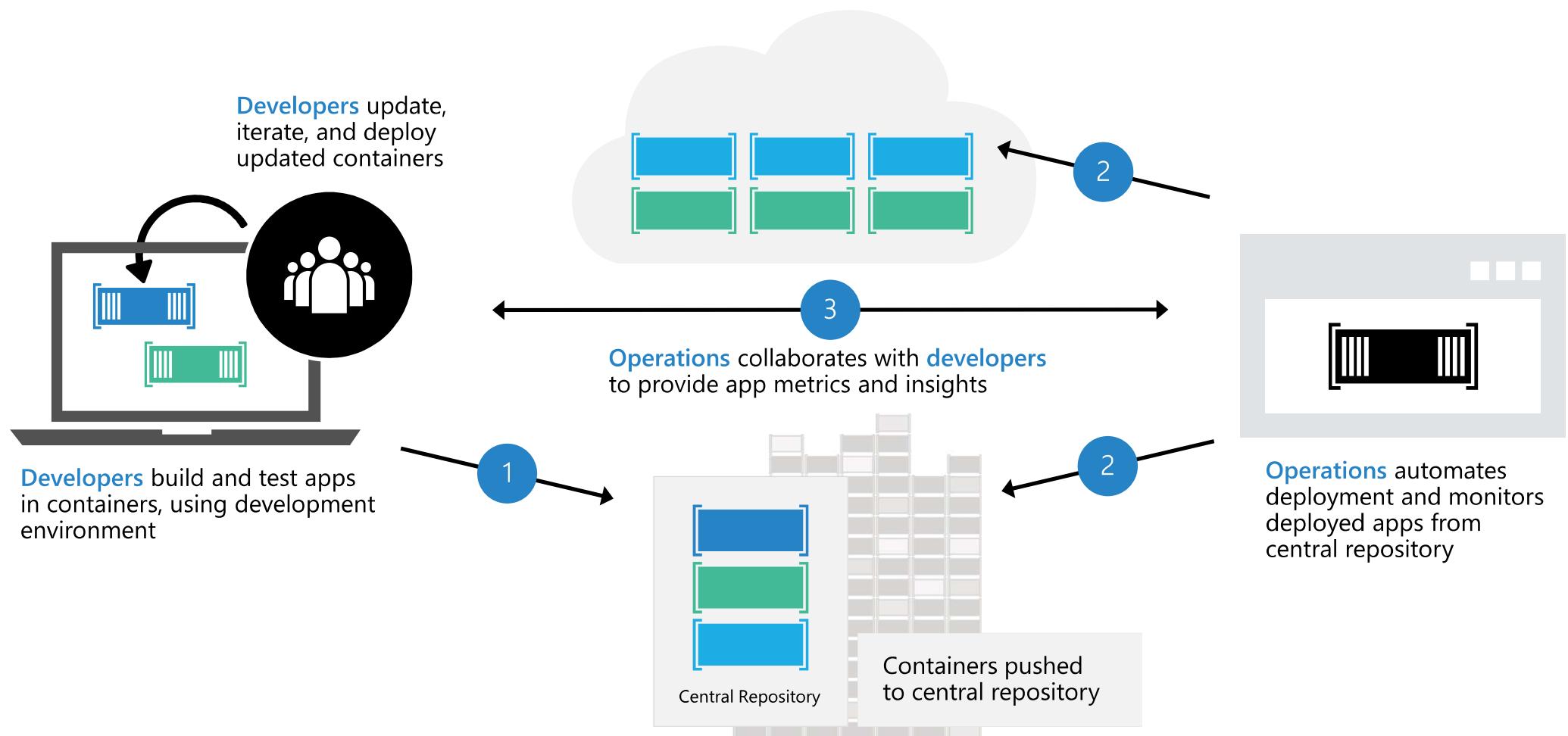


DEMO

Docker Datacentre

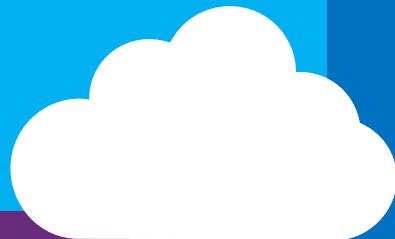
Docker in Release Pipeline

DevOps process with containers





Visual Studio
Team Services



Team Foundation
Server



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>



Open + Flexible Tool Chain

1,177
Visual Studio
Code Extensions

6,538
Visual Studio
Gallery Extensions

90
Visual Studio
Sim-Ship Partners

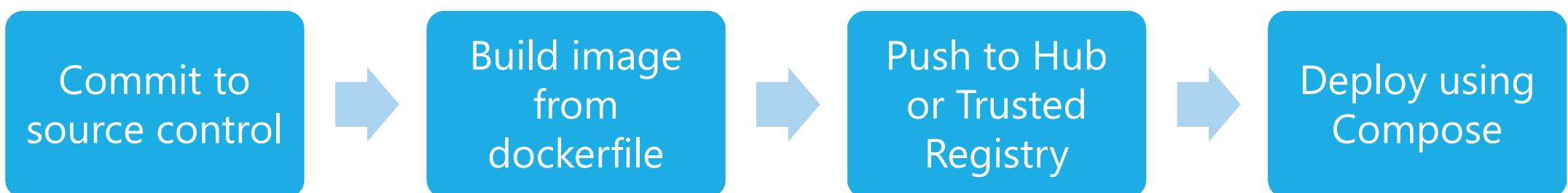
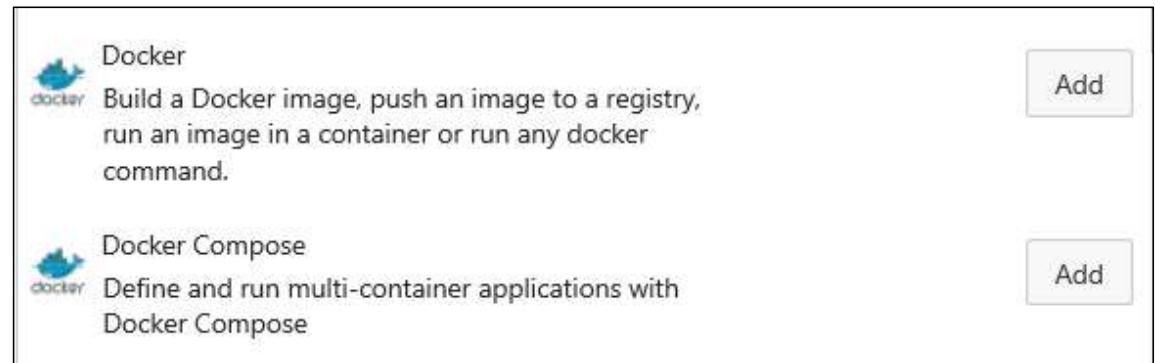
171
VS Team Services
Extensions



Visual Studio Team Services

- Docker and Docker Compose tasks available for build and release:

- Build an image
- Push an image
- Run an image
- Generic Docker command
- Deploy a compose-file



DEMO

VSTS Build & Release to
Docker Datacentre

Want to try this out?

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

			
Developer tools	Cloud services	Software	Training and support
Editors, designers, and debuggers to develop for any platform	Compute, storage, analytics, team collaboration and more	Trials and downloads, from operating systems to Office online apps	World-class technical training and priority support

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

		
Cloud Services	Education	Support
Get in the cloud with Azure and extended trials of Office 365 and Enterprise Mobility Suite	Increase your cloud knowledge and get certified with training courses from Microsoft Virtual Academy and PluralSight	Get a free support incident and priority support in TechNet forums

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>

Contact me:

marrobi@microsoft.com
@techdiction

Questions?