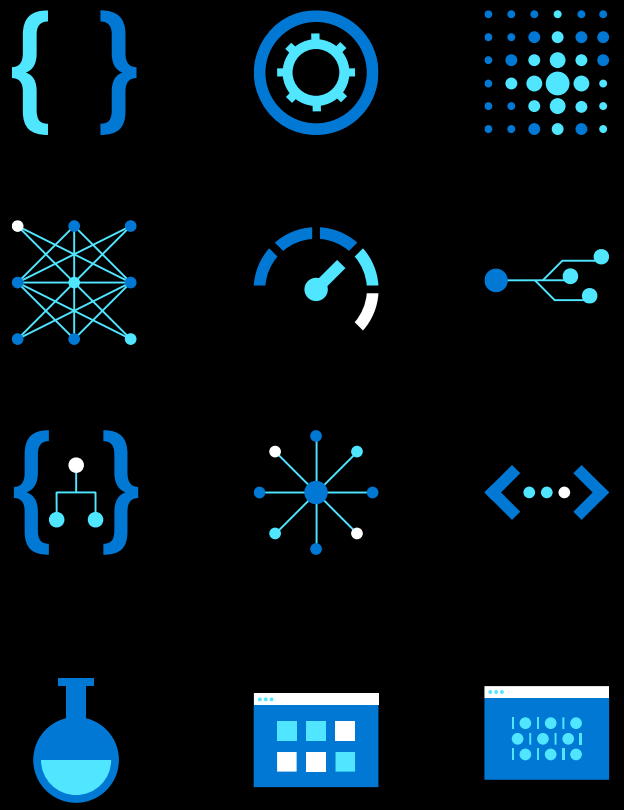
Microsoft Azure

Azure Training Day

Run cloud-native apps with Azure Kubernetes Service



Microsoft

Azure

Run Docker containerized workloads on Azure

Part 1 of 4 in the Run cloud-native apps with Azure Kubernetes Service series

MSUSDev@Microsoft.com

Oct, 2019

About us…

David Hoerster

- Sr. Cloud Solution Architect

Matthew Calder (macalde@microsoft.com)

- Sr. Product Marketing Manager – Azure, US Marketing and Operations

For questions or help with this series
MSUSDev@Microsoft.com

For the lab guides and sample code
https://github.com/MSUSDEV/Run-Cloud-Native-Apps-With-AKS

Setting the scene

Overview of the workshop

About the workshop content…

About:
This series is the second half of a longer workshop that teaches how to build a proof of concept (POC) that will transform an existing ASP.NET-based Web application (SimplCommerce) to a container-based application. You can register to view the modules from the first half at https://aka.ms/web-app-series You can find all the presentations form the first half at https://github.com/MSUSDEV/Migrating-web-apps-to-Azure

At the end of this workshop, you will have a good understanding of container concepts, Docker architecture and operations, Azure Container Services, Azure Kubernetes Services and Azure DevOps tools.

Target Audience:
The workshop is targeted to Cloud Architects, Cloud Solution designers, developers and IT sysadmins, CIO's, CTO's and anybody else who is interested in learning about Azure, containers, application cloud migration and digital transformation.
Focus of the workshop (40%) is getting hands-on experience, complemented with presentations and whiteboard sessions (if in-person delivery).

Time Estimate:
11 hours (+/- 5 hours presentations, 6 hours of optional hands-on labs for attendees)

Workshop Agenda - Presentations

What we will talk about…

Series 1: https://aka.ms/web-app-series

- Module 1: Digital App Transformation with Azure
- Module 2: Running Azure Infrastructure and execute Lift & Shift Migrations
- Module 3: Performing proper assessments to smooth Azure Migrations
- Module 4: Why and how migrating databases to Azure PaaS
- Module 5: Migrating to Azure App Services – Azure Web Apps (.NET)

Series 2: https://aka.ms/cloud-native-series

- Module 1: Deploying Containers on Azure (YOU ARE HERE)
- Module 2: Deploying Azure Kubernetes Services
- Module 3: Introduction to Azure DevOps
- Module 4: Optimizing Azure Operations and Monitoring

Workshop Agenda – Hands On Labs

From series 1

- **Module 2: Running Azure Infrastructure and execute Lift & Shift Migrations**
- *Lab 1: Deploy an Azure VM Infrastructure using ARM-Templates*

- **Module 3: Performing proper assessments to smooth Azure Migrations**
- *Lab 2: Using Azure assessment tools*

- **Module 4: Why and how migrating databases to Azure PaaS**
- *Lab 3: Migrating SQL Databases to Azure using Database Migration Assistant*

- **Module 5: Migrating to Azure App Services – Azure Web Apps (.NET)**
- *Lab 4: Publishing application source code to Azure Web Apps using Visual Studio 2019*

Workshop Agenda – Hands On Labs

For this series 2

- **Module 1: Deploying Containers on Azure**
- *Lab 5: Containerizing applications using Docker and running it in Azure Container Instance and Azure WebApp for Containers*

- **Module 2: Deploying Azure Kubernetes Services**
- *Lab 6: Deploying Azure Kubernetes Services and running containerized apps from Azure Container Registry*

- **Module 3: Optimizing Azure Operations and Monitoring**
- *Lab 7: Monitoring and Managing your Azure deployed workloads*

- **Module 4: Introduction to Azure DevOps**
- *Lab 8: Deploying Azure DevOps with CI/CD Pipelines and deploy your applications to Azure WebApps, WebApp for Containers, Azure Container Instance and Azure Kubernetes Services*

Technical Requirements

What you need…

- See appendix slides for lab dependencies and / or alternate path for workshop
- Client workstation running recent Windows, Linux or Mac OS and latest internet browser
- Access to ports 80 (HTTP), 443 (HTTPS) and 3389 (Remote Desktop)
- Full Azure subscription (MSDN, AzurePass, Paid subscription, AE, CSP,…), where you have Owner permissions on subscription level
- Lab consumption estimate: $15-35

Questions and HOL support

For questions or help with this series
MSUSDev@Microsoft.com

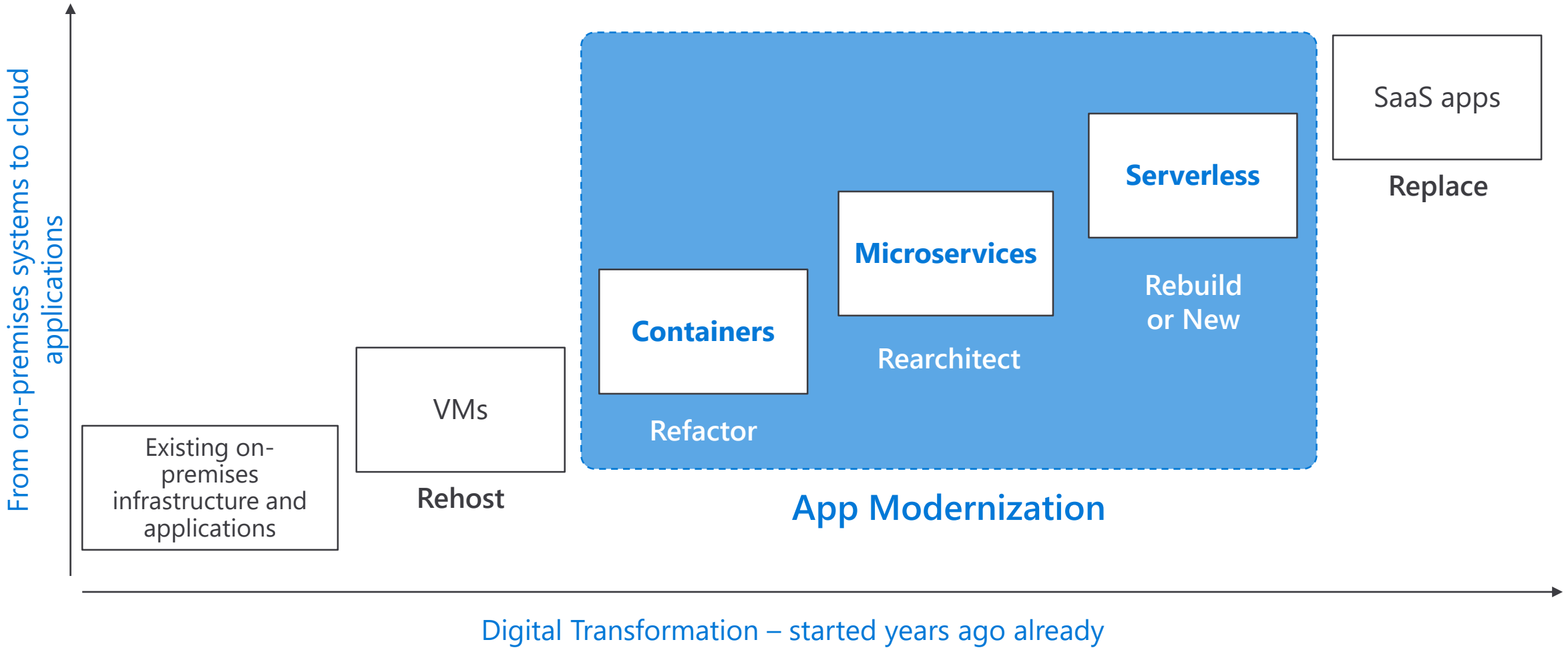
For the lab guides and sample code
https://github.com/MSUSDEV/Run-Cloud-Native-Apps-With-AKS

For information about lab dependencies and alternate approach please see the appendix slides at the end of this presentation.

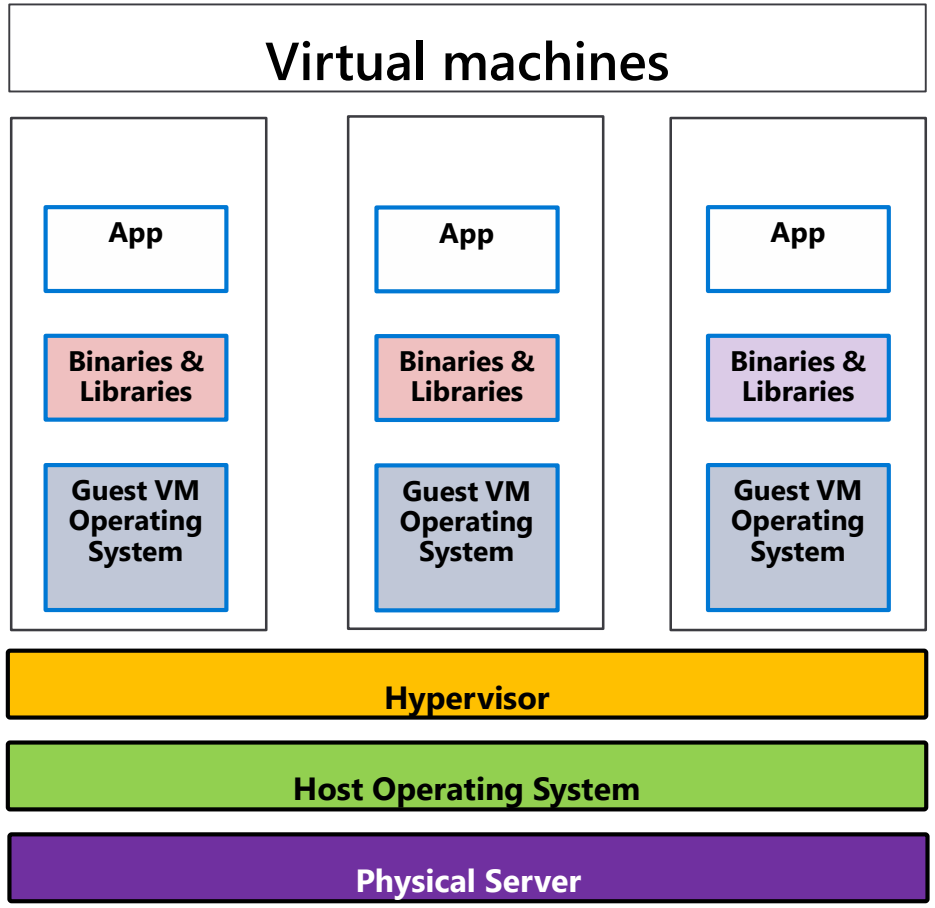
Introduction to Containers

The journey to the cloud

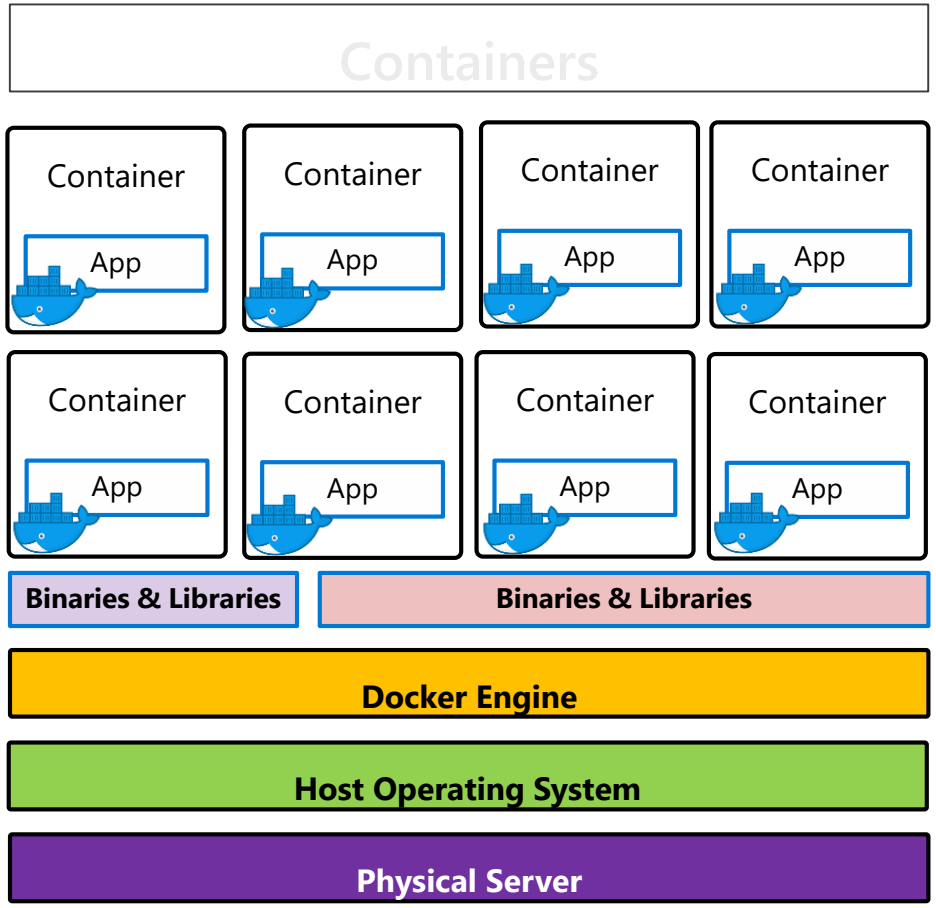
From Virtual Machines to Software as a Service



What are containers?



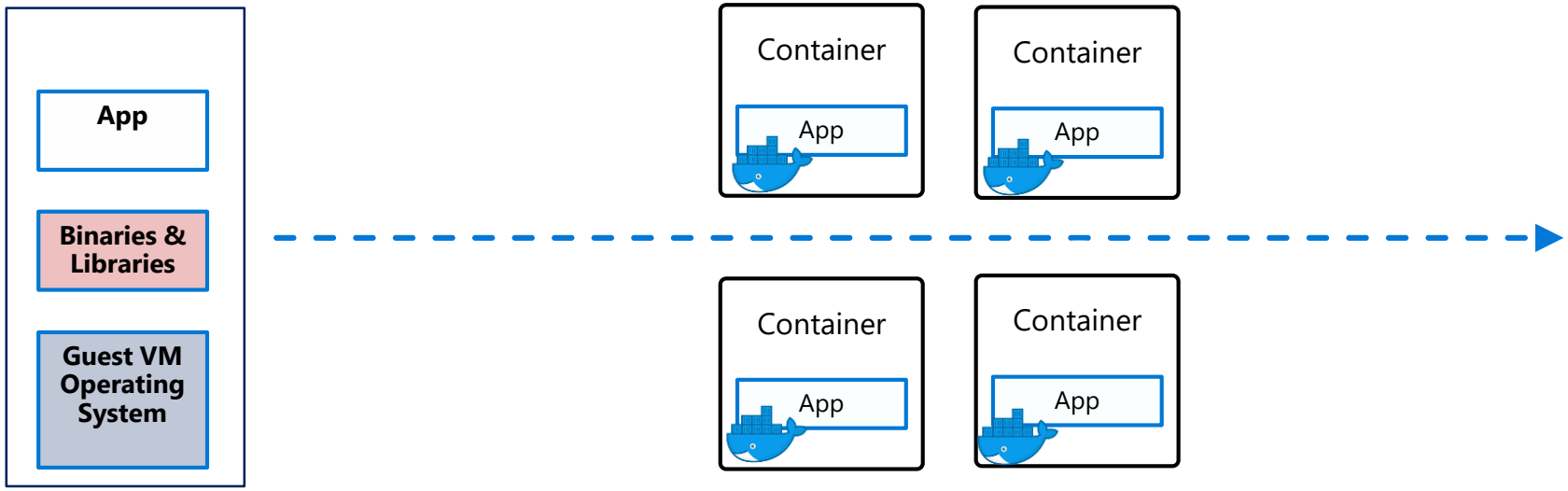
- Virtualize the **hardware**
- **VMs** as units of scaling
- Hypervisor **dependent**
- **Not** easily movable



- Virtualize the **operating system**
- **Applications** as units of scaling
- Platform **independent**
- **Easily** movable across environments (on-premises, multi-cloud)

How do containers help in app modernization?

Containers are stand-alone, smaller application instances, running at scale



Existing/New Applications

(=rehome)

Lift and shift to containers

(= rehost, refactor)

Running Containers on Azure: a full set of choices

App Service

Deploy web apps or APIs using **containers** in a PaaS environment

Service Fabric

Modernize **.NET applications** to microservices using **Windows Server containers**

Kubernetes Service

Scale and orchestrate **Linux containers using Kubernetes**

Container Instance

Elastically burst from your **Azure Kubernetes Service** (AKS) cluster

Partner Ecosystem

Bring your **Partner solutions** that run great on Azure

Azure Container Registry

Docker Hub

Choice of developer tools and clients

How Containers help in App Migration

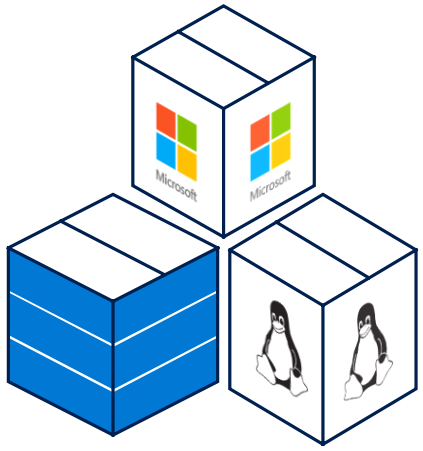
Enterprises can't get away from their legacy apps just like that:

- Expensive
- Risk involved
- Developers are gone
- Complexity

Enterprises benefit from Containers:

- Legacy apps are supported
- Containers are future-proof
- Cost optimized
- Secure

How Containers help in App Migration

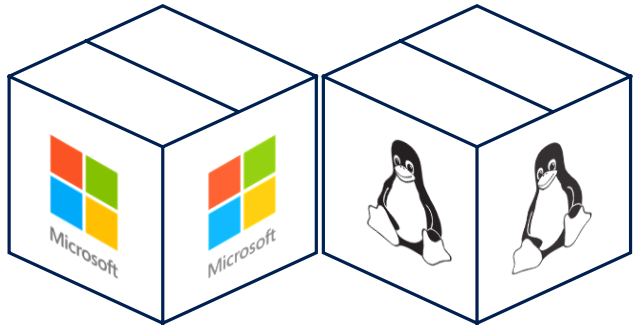


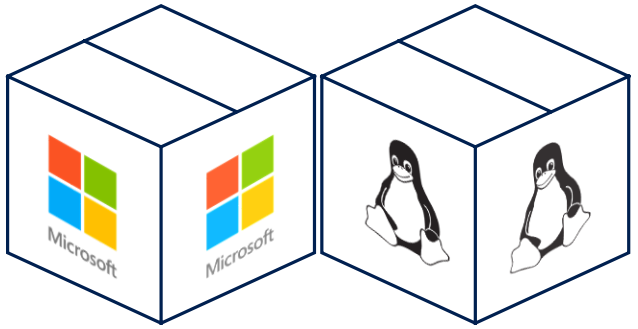
Container Image
Windows or Linux

Run
Anywhere

On-Premises

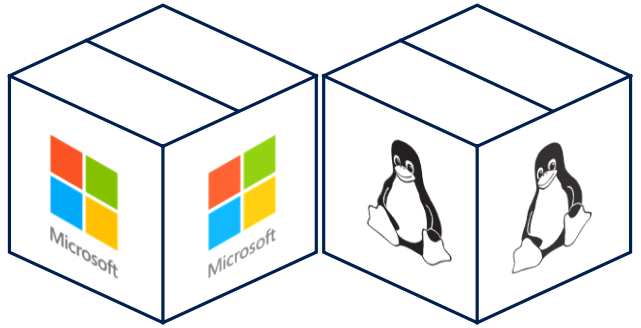
Co-location

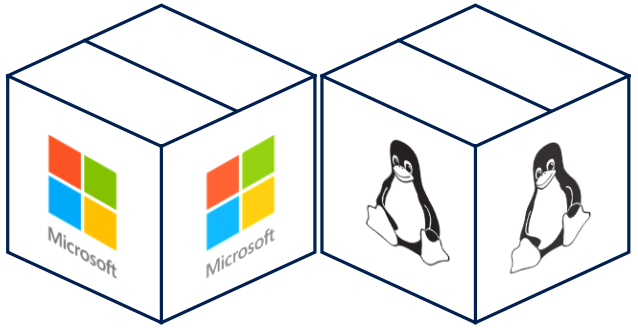




Public Cloud XYZ

Azure



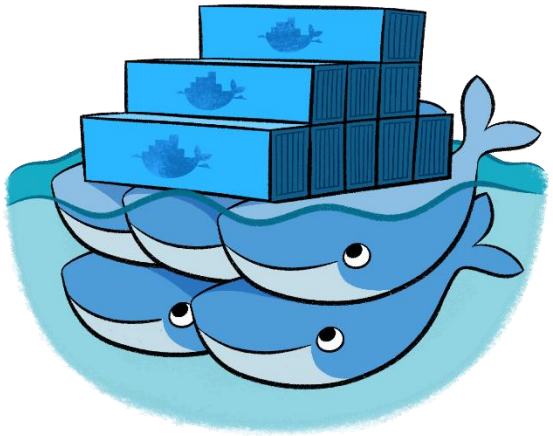


Docker Containers Overview

Docker Containers - Overview

What is Docker?

- Leading Open-Source Containerization Platform
- Natively Supported in Azure
- Cross-Platform (Win, Linux,…)



Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in

Source: www.docker.com

Docker High-Level Architecture

Docker Concepts

1

Client

Docker build

Docker pull

Docker Run

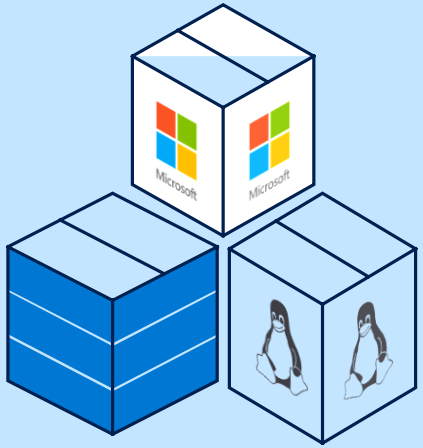
Docker …

2

Host

Docker Daemon / Docker Service

Containers



Images

3

Registry

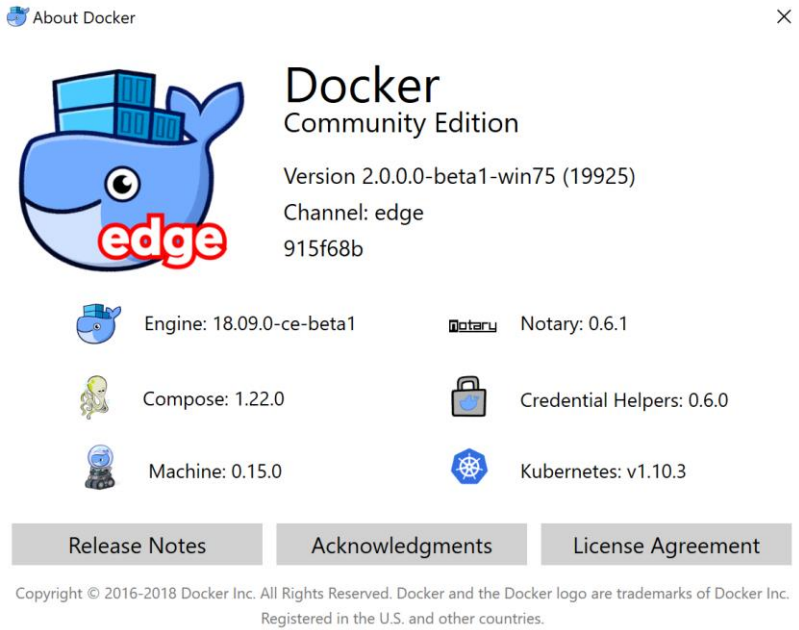
Docker Hub

Azure Container Registry

Cloud XYZ Container Registry

Docker Host

- Windows-based (Win10/Win2016-2019)
- Linux-based

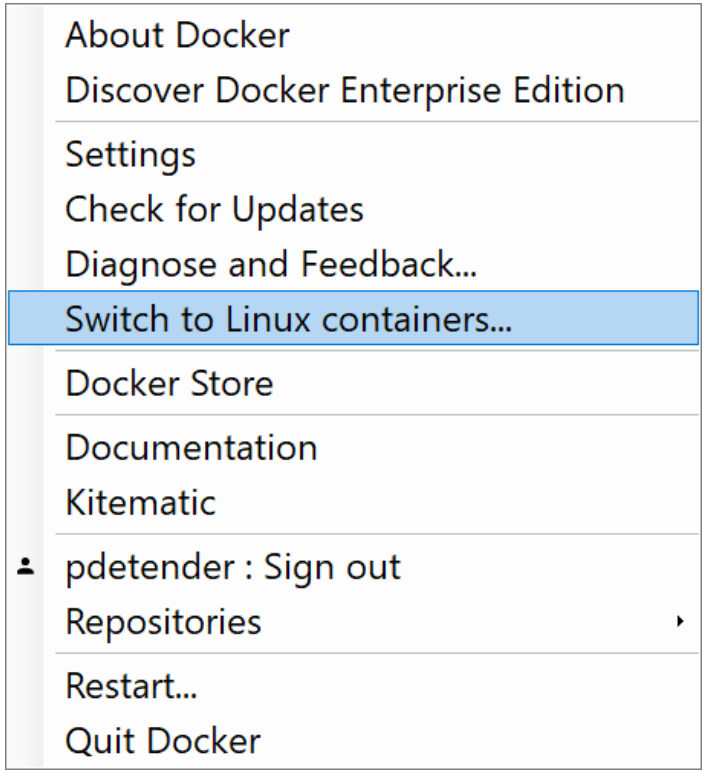



```
sudo docker run -it microsoft/azure-cli
d image 'microsoft/azure-cli:latest' locally
ng from microsoft/azure-cli

Pull complete
Pull complete
Pull complete
Pull complete
5:63122c945fc648333b5ae633dc532ef4d6c3c3cf474a945a9fb0b50bd4c753e9
oaded newer image for microsoft/azure-cli:latest
e5f1:/#
```

Docker Containers on Windows

- Runs on Windows 10 client or Windows Server 2016/2019
- Supports both Windows and Linux Containers, and you can easily switch
- Requires Hyper-V or « Containers » Feature
- CLI integration with PowerShell or command prompt
- Docker CE = Free !!



Docker CLI

Command Line Interface

Command Prompt

```
C:\Users\P>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\P\\.docker")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\P\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\P\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\P\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  builder     Manage builds
  config      Manage Docker configs
  container   Manage containers
  engine      Manage the docker engine
```

Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> docker pull microsoft/azure-cli
Using default tag: latest
latest: Pulling from microsoft/azure-cli
693502eb7dfb: Pull complete
995988fe2b30: Pull complete
4ef74641d502: Pull complete
a18d90d86a2e: Pull complete
Digest: sha256:d3ee5a39ee681389e25ffe9fc4413d388bee34b5477bb0b47a1c24c6d9beb60e
Status: Downloaded newer image for microsoft/azure-cli:latest
PS C:\WINDOWS\system32> docker run -it microsoft/azure-cli
```

```
ubuntu:~$ docker version
nt:
sion:      1.11.2
 version:  1.23
version:   go1.6.2
 commit:   b9f10c9
lt:        Thu, 16 Jun 2016 21:17:51 +1200
Arch:      linux/amd64
ot connect to the Docker daemon. Is the docker
```

Demo

Running Docker for Windows

Docker Host

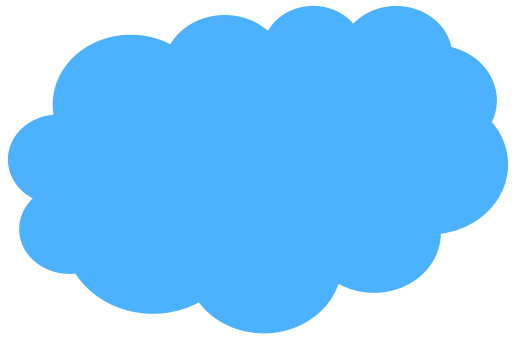
Where do you run your Docker Container Images?

On top of a « Docker » cloud platform:

- Azure Container Instance
- Azure Container Services (AKS)
- Azure Virtual Machines

On top of a « host » Operating System, like:

- Windows 10
- Windows Server 2016/2019
- Linux
- Mac OS



Where do Docker Images come from?

Docker Hub

- Hub.docker.com
- Free resource of PUBLIC images
- Option to create PRIVATE images

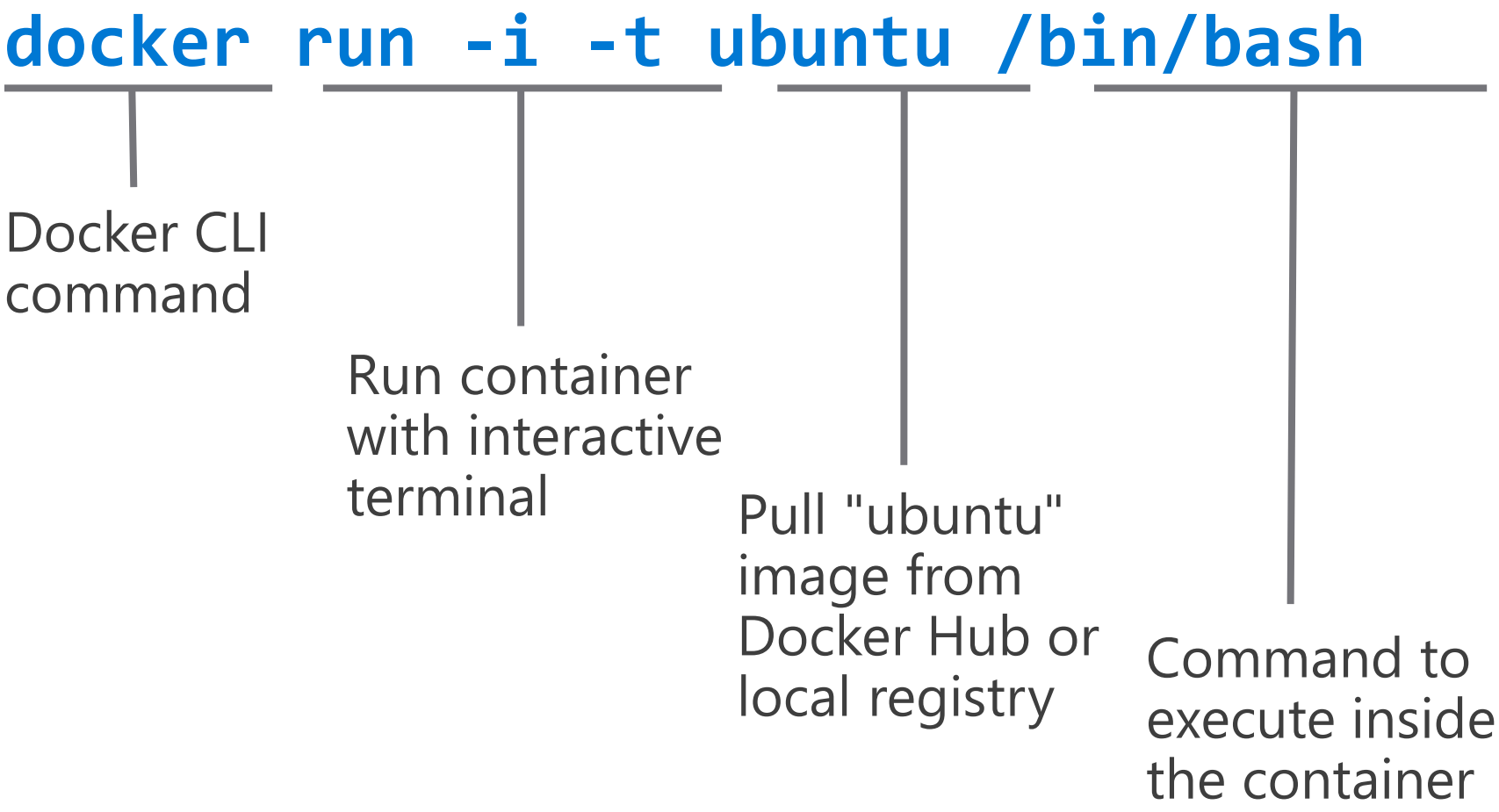
Docker Hub

Cloud Container Registry

- Library of Docker Images
- Azure Container Registry
- Mainly used for storing PRIVATE images

Azure Container Registry

Running a Docker Container



Common Docker CLI operations

`docker run`	- Use an image to run a container
`docker pull`	- Pull an image from a registry
`docker build`	- Build a Docker image
`docker images`	- List available Docker images
`docker ps`	- List running Docker containers
`docker exec`	- Execute a command in a container
`docker stop`	- Stop a running container

Demo

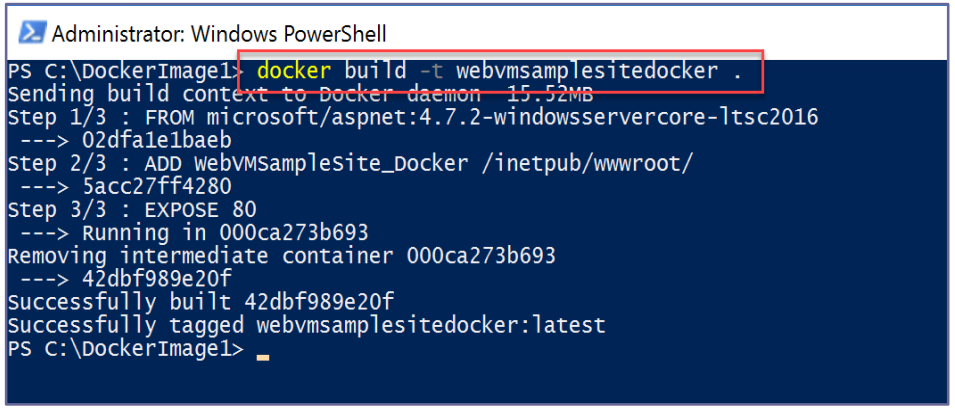
Common Docker CLI Operations

Migrating apps to Docker Images

Building a Docker Image - CLI

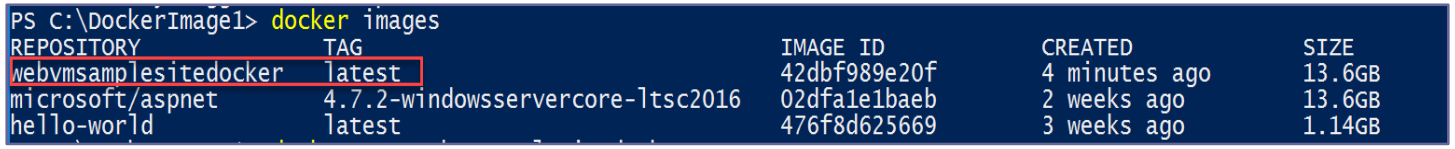
1 **`docker build`**

- Build a Docker image



2 **`docker images`**

- List available Docker images



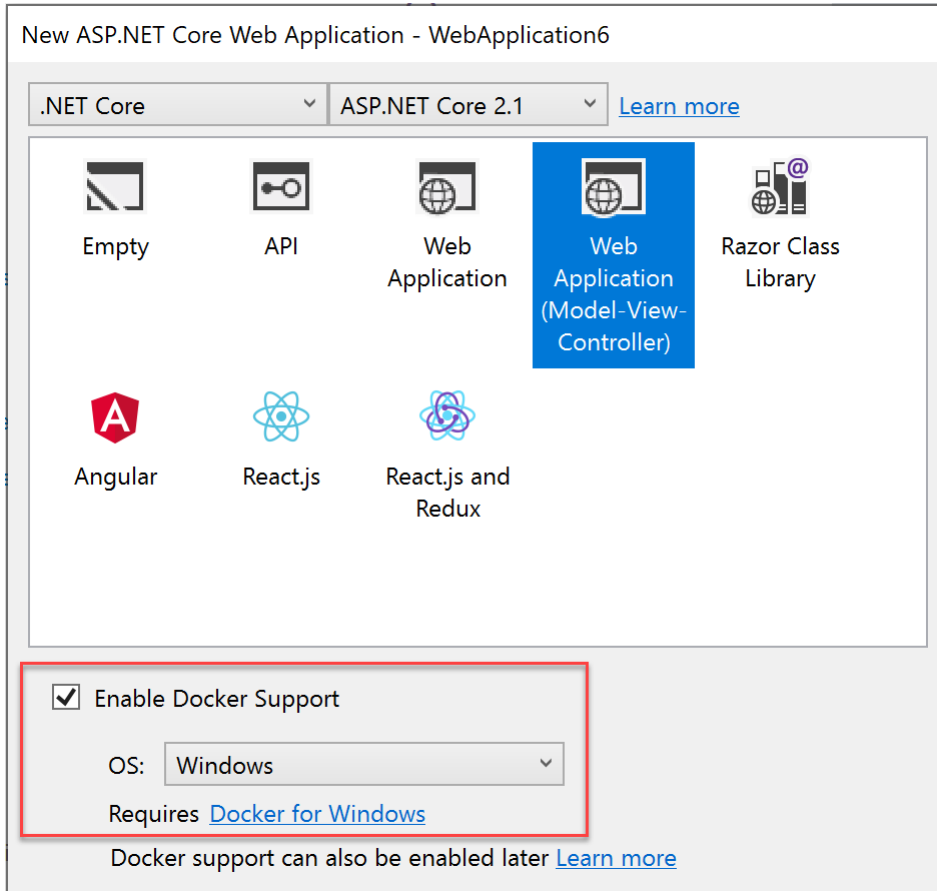
Building a Docker Image - DockerFile

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using **`docker build`**, users can create an automated build that executes several command-line instructions in succession.

```
FROM microsoft/aspnet:4.7.2-windowsservercore-ltsc2016
ADD WebVMSampleSite_Docker /inetpub/wwwroot/
EXPOSE 80
```

Building a Docker Image – VS2017-2019

When using Visual Studio 2017-2019, together with Docker for Windows on the same client, you get Docker integration features



https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/docker/visual-studio-tools-for-docker?view=aspnetcore-2.1

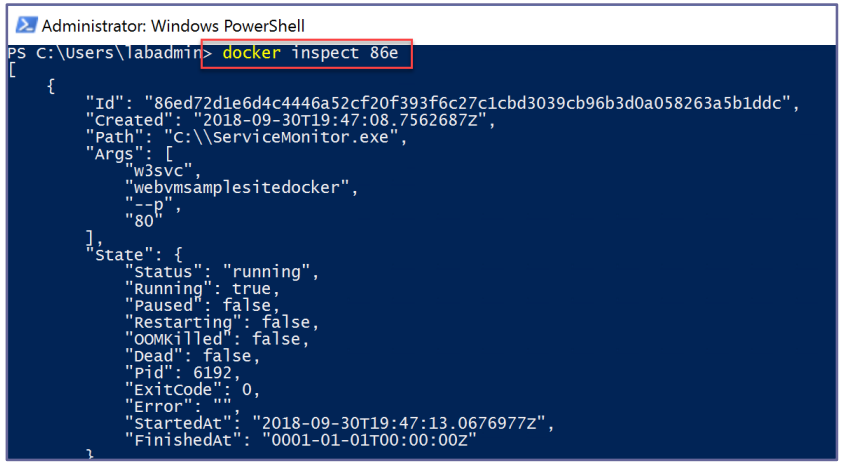
Demo

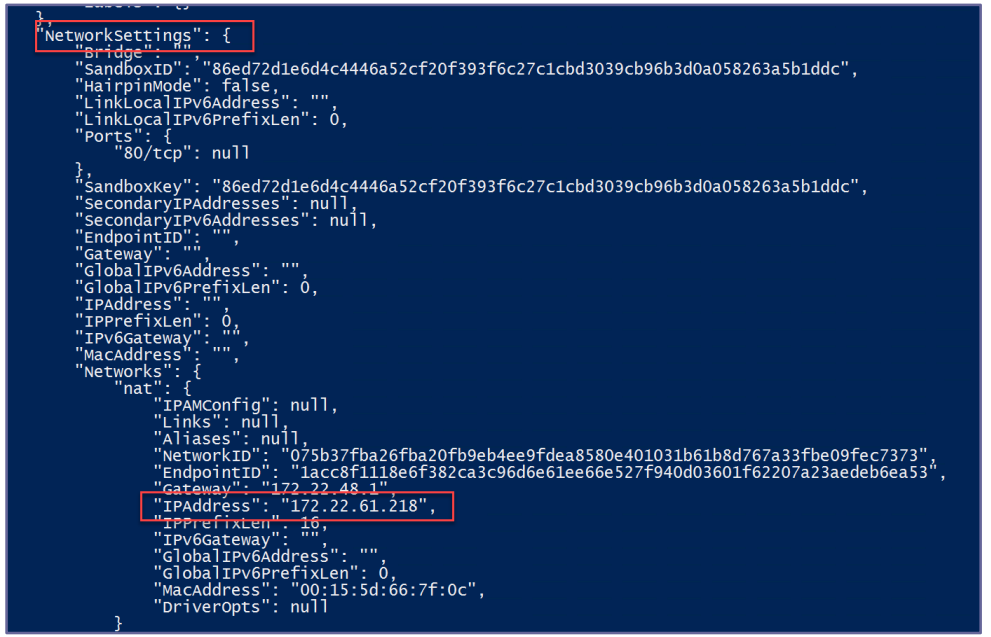
Building a Docker Image (CLI and VS2019)

Troubleshooting Docker Containers

1. **`docker container ls`**
   – Lists all containers on a host

2. **`docker inspect`**
   – Shows all information of a container





Azure Container Registry

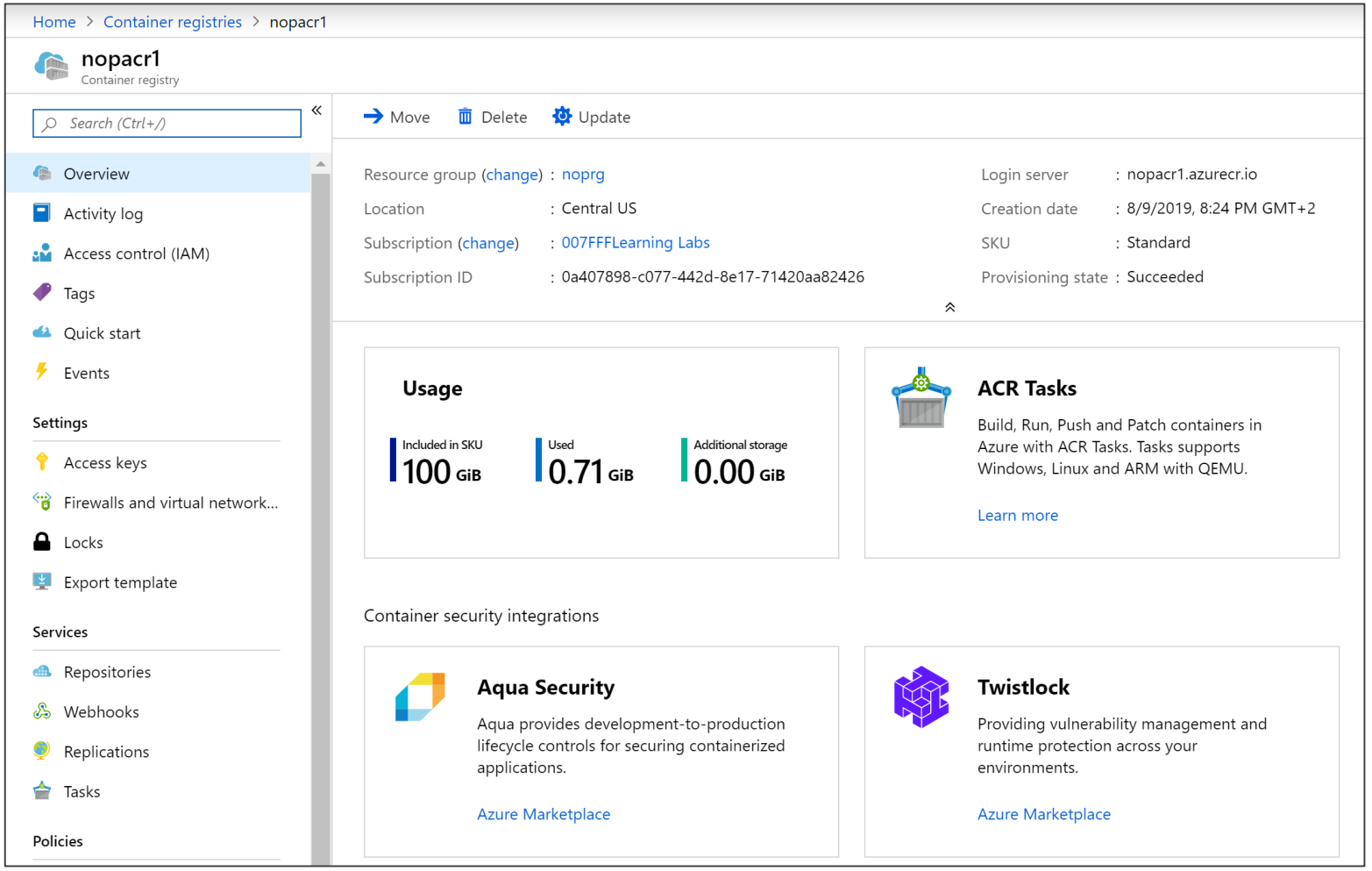
Azure Container Registry (ACR)

Azure Container Registry is a managed Docker registry service based on the open-source Docker Registry 2.0. Create and maintain Azure container registries to store and manage your private Docker container images.

- **Pull images from ACR and use it in different deployment targets:**
- Azure Kubernetes Services (AKS)
- DC/OS MesoSphere
- Docker Swarm
- Azure compute solutions
- Azure WebApp for Containers
- Azure Container Instance (ACI)

- **3 different SKU's:**
- Basic
- Standard
- Premium

Azure Container Registry (ACR)



Similar to the public Docker Hub, but running within an Azure Subscription, offering Private images

Azure Container Registry (ACR)

Automating OS and Framework Patching

Registry

Registryname.azurecr.io
namespace

Image

Read-only snapshop of a
Docker Container

Repository

Groups of Container
Images

Container

Software application in a
bundled file system

Azure Container Registry (ACR)

3 different SKUs to choose from

Basic	Standard	Premium
Cost optimized entry-point for developers	Sufficient for most production workloads	For the enterprise-use of containerized apps
Same capabilities than Standard and Premium, but limitations on size and usage	Increased storage limits and image througput	Higher offerings on storage, concurrent operations and high-volume scenarios
		Geo-Replication for managing a single registry across multiple regions

Azure Container Registry Tasks

Cloud-based container building, including OS updates and patching

- Quick Tasks – container lifecycle management
- Trigger container image build when updates happen to Git Repo
- Automate OS and Framework patching (base image)
- Automatically rebuild application images (app image)
- Multi-Step Tasks (preview)

Demo

Deploying an Azure Container Registry

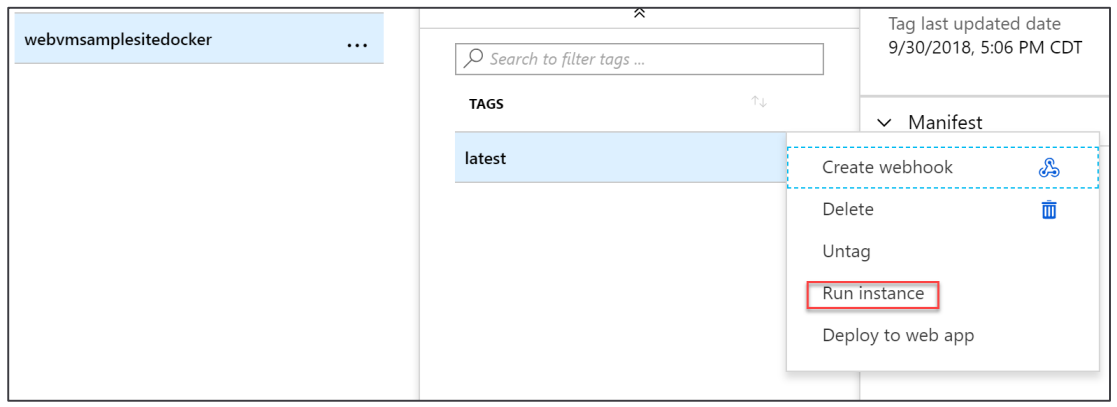
Azure Container Instance

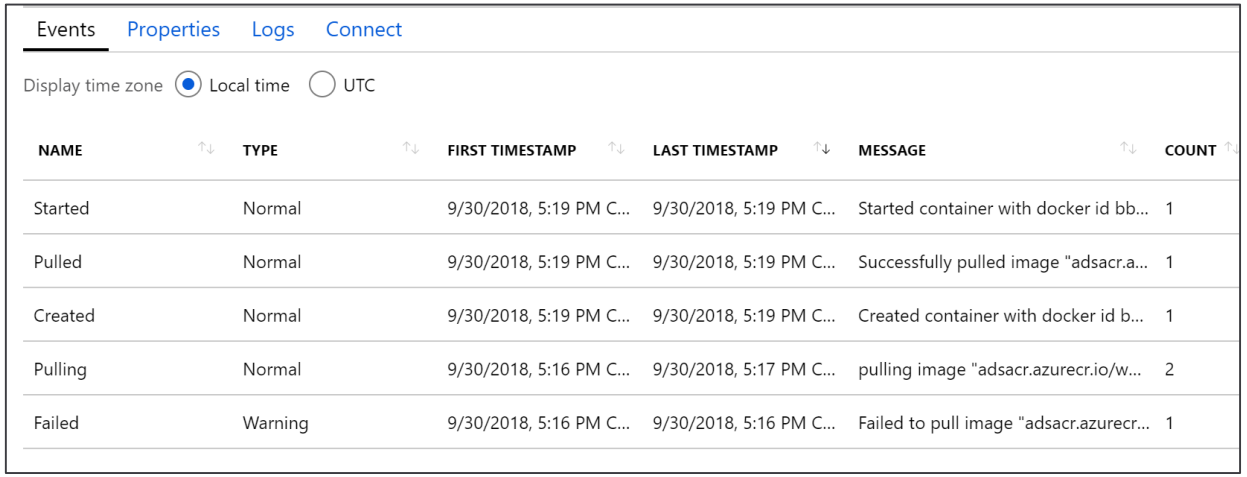
Azure Container Instance (ACI)

Azure Container Instances offers the fastest and simplest way to run a container in Azure, without having to provision any virtual machines and without having to adopt a higher-level service.

Run Containers directly from Azure Container Registry:

- Detailed Event Logging
- No additional Orchestration needed
- Integrates with Azure monitoring and diagnostics capabilities





NAME	TYPE	FIRST TIMESTAMP	LAST TIMESTAMP	MESSAGE	COUNT
Started	Normal	9/30/2018, 5:19 PM C...	9/30/2018, 5:19 PM C...	Started container with docker id bb...	1
Pulled	Normal	9/30/2018, 5:19 PM C...	9/30/2018, 5:19 PM C...	Successfully pulled image "adsacr.a...	1
Created	Normal	9/30/2018, 5:19 PM C...	9/30/2018, 5:19 PM C...	Created container with docker id b...	1
Pulling	Normal	9/30/2018, 5:16 PM C...	9/30/2018, 5:17 PM C...	pulling image "adsacr.azurecr.io/w...	2
Failed	Warning	9/30/2018, 5:16 PM C...	9/30/2018, 5:16 PM C...	Failed to pull image "adsacr.azurecr...	1

Demo

Running an Azure Container Instance

Webapp for Containers

Running Containers as WebApps

Classified as Microsoft Confidential

Web App now supports running Containers

Bring your code

Bring your container

Web App for Containers

- ✓ **Deploy to Azure in seconds**
- ✓ **Scale easily on demand**
- ✓ **Designed for your agile web development needs**

Containers as Web Apps - Benefit

Why running Containers as Azure Web Apps

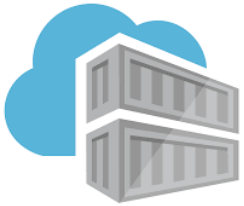
- Treat the container as a web app
- All common Azure Web App features are valid for Containers as well:
- Backup
- Monitoring
- App Insights
- App Service Plan (Although the plans are different for containers!)

Azure App Service: Build and Deploy Options

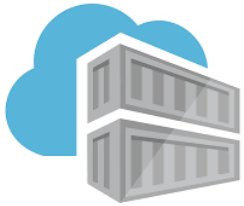
Deploying containers

Azure Pipelines

Docker push



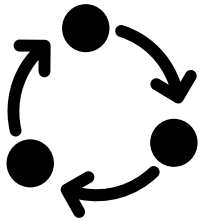
Private container registry



Azure Container Registry

Pulls container image

Runnable web app
in Azure App Service



Docker build:

- Docker Hub build
- Docker build on dev machine
- Etc…

Docker Hub

Web app for containers

High productivity development

Deployment with ease

CI/CD build and deploy

Testing in production

Staged deployment with slots

Fully managed platform

Built-in auto scale and load balancing

High availability with auto-patching

Monitoring and diagnosis

Backup and recovery

Enterprise-grade apps

Global data center footprint

Private registry support

AAD integrated

Secure + compliant

Windows Container Support Public Preview

Windows Server 2019 Host Support

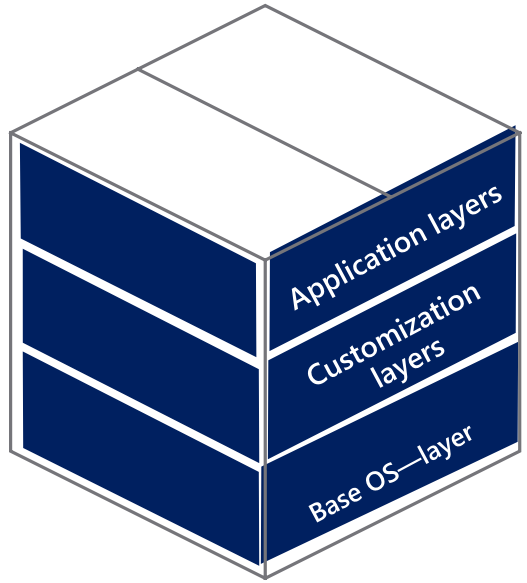
- Smaller containers, higher density of apps, faster pull and start times
- Server Core Containers reduced by over 60% from WS2016LTSC
- Take advantage of improvements not available on WS2016

Key Scenarios

- Lift and Shift to PaaS
- Applications which have dependencies
- Applications blocked by traditional App Service Sandbox
- Data center migration

Capabilities

- Available in six regions globally
- PowerShell and CLI Support
- Bring Your Own Storage – Azure Files



Demo

Deploying Azure WebApps for Containers

Section Take-Aways

1. Docker is the standard for containerized applications

2. Containers allow for digital transformation, by supporting your (legacy) applications to modern platforms, both on-premises and in the cloud

3. Azure fully supports running containers using Azure Container Registry, Azure Container Instance, WebApp for Containers and Azure Kubernetes Services

Questions Landing Spot

"…If you want good answers, ask better questions…"

© Randy Glasbergen

Next Module...

Deploying and Running Azure Kubernetes Services

Appendix

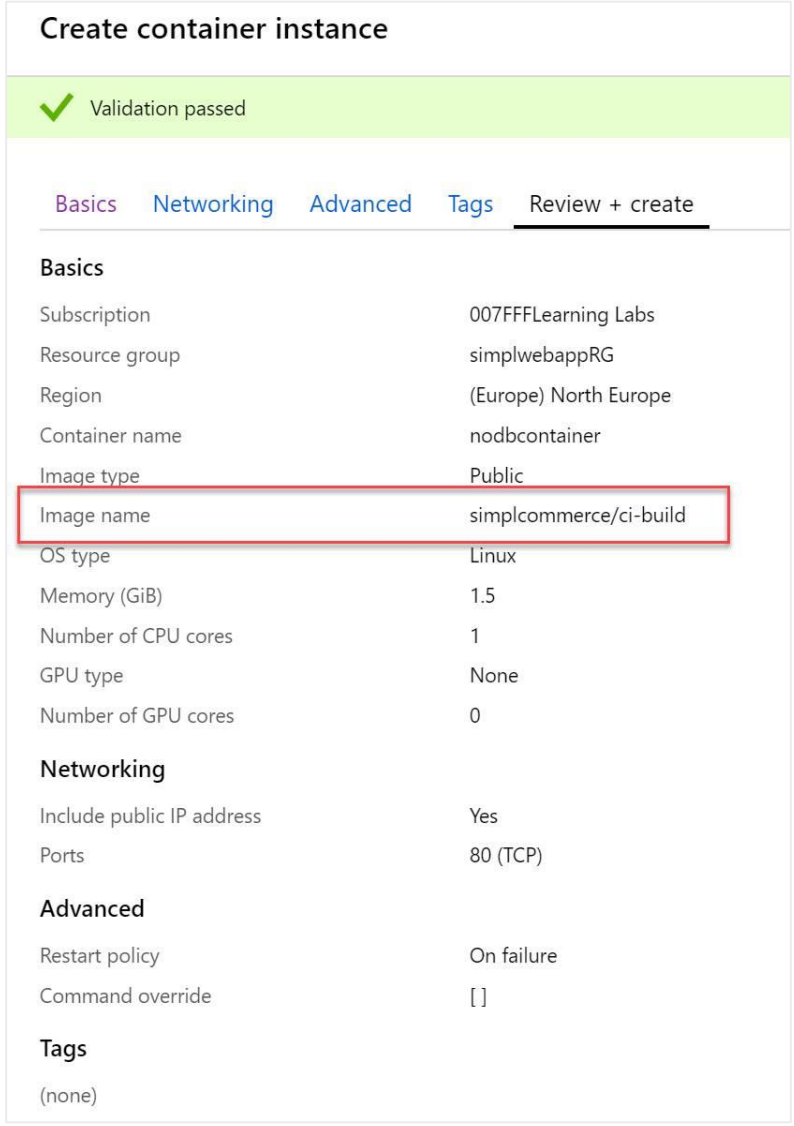
Lab dependencies and alternate path

Classified as Microsoft Confidential

Alternate path

To avoid the SQL dependencies from the first series labs

- SimplCommerce offers a Docker container that uses a built-in non-SQL database
- https://hub.docker.com/r/simplcommerce/ci-build
- If you run this locally (or in ACI or AKS) it will spin up the web app and give you the option to select “sample products” (phones or fashion)



Full workshop

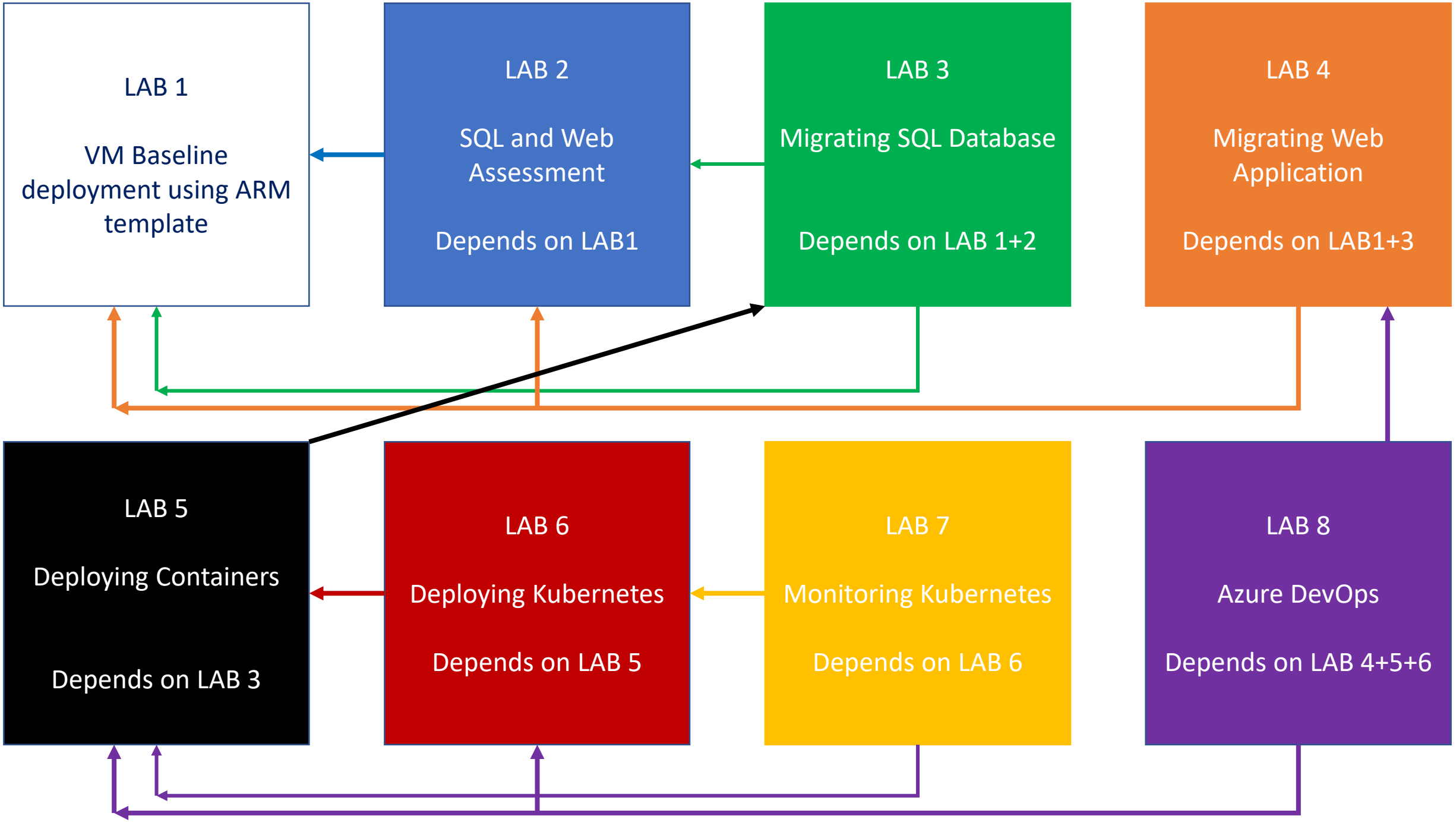
Dependencies from first series

The lab guide assumes that the learner has completed the first 3 labs from the previous series “Migrate a web app to Azure”

Deploying VM baseline using ARM Templates

Performing Assessments

Migrating SQL Databases


LAB 1
VM Baseline deployment using ARM template
LAB 2
SQL and Web Assessment
Depends on LAB1
LAB 3
Migrating SQL Database
Depends on LAB 1+2
LAB 4
Migrating Web Application
Depends on LAB1+3
LAB 5
Deploying Containers
Depends on LAB 3
LAB 6
Deploying Kubernetes
Depends on LAB 5
LAB 7
Monitoring Kubernetes
Depends on LAB 6
LAB 8
Azure DevOps
Depends on LAB 4+5+6

Microsoft

Thank You

© Copyright Microsoft Corporation. All rights reserved.