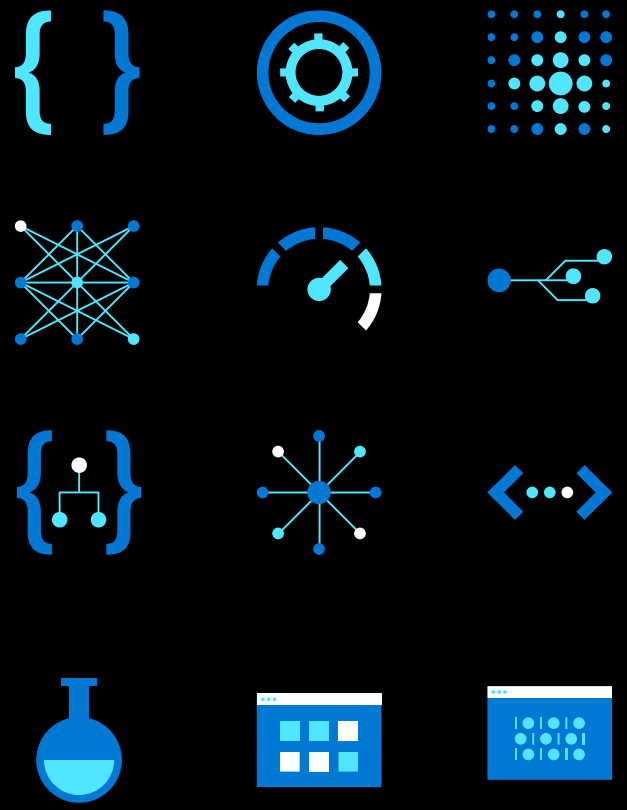
Microsoft Azure

Azure Training Day

Migrating web applications to Azure



Microsoft

Azure

Overview of Azure web app migration

Part 1 of 5 in the Migrate web apps to Azure series

MSUSDev@Microsoft.com

Sep, 2019

About us…

Fareed Muhammad Khan

- National Solutions Director – Azure, US Regulated Industries

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/Migrating-web-apps-to-Azure

Setting the scene

Overview of the workshop

About the workshop content…

About:
This series, Migrate web apps to Azure, takes you through the first 5 modules of a longer, 9 part workshop that builds a proof of concept (POC) that transforms an existing ASP.NET-based Web application (SimplCommerce) to a container-based application.

For the purpose of this training, we will focus on the steps that take you through deploying and migrating web applications to Azure Web Apps: automating Azure resources deployments using Visual Studio and Azure Resource Manager (ARM) templates, performing proper assessments and the tools Microsoft offers to help in this migration preparation phase, Microsoft SQL database migration to SQL Azure PaaS, as well as deploying and migrating web applications to Azure Web Apps.

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 webcast presentations.

Time Estimate:
9 hours (+/- 6 hours presentations, 3 hours of optional hands-on labs for attendees)

Workshop Agenda - Presentations

What we will talk about…

- 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)

Workshop Agenda – Hands On Labs

Learn by doing…

- **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*

Technical Requirements

What you need…

<Could vary based on the actual delivery-method>, but overall:

- 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

MSUSDev@microsoft.com

Subject: Azure Developer Series – migrate web apps

Response Time: within 24 hours

Check GitHub for FAQ and Updates:
https://github.com/MSUSDEV/Migrating-web-apps-to-Azure

Questions Landing Spot

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

© Randy Glasbergen

Why and How does Azure support digital transformation

Digital Transformation

Key Objectives

What you will learn in this section

- Digital Transformation
- The journey to the cloud
- Introduction to Containers
- Introduction to Microservices
- Technologies covered in this workshop series

Digital Transformation

What is this

digital transformation

everybody talks about?

Digital transformation helps organizations to **improve efficiency and effectiveness** of service/products, using the **latest available technology**.

Instead of replicating an existing platform, we use technology to **transform that service/product into something significantly better.**

Digital Transformation

What is

digital transformation

for IT Admins?

Digital Transformation is using technology to **radically change the datacenter**

Don't focus on **the location of the technology**, but rather focus on **the use of technology**

Embrace the **continuous flow** of updates and changes, and **keep up** with it.

Digital Transformation

What does

digital transformation mean

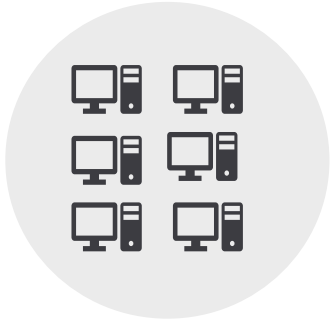
for Developers?

Digital Transformation is using technology to **radically change your application landscape**

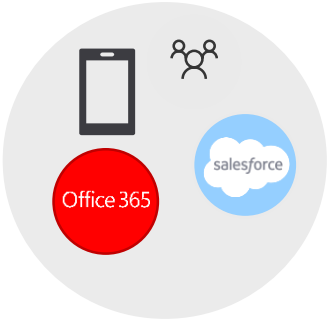
Don’t focus on **dev language**, but rather focus on **the capabilities a language brings**

Importantly, think **broader** than just the application,the data,… but how it runs and scales

Digital Transformation is continuously evolving

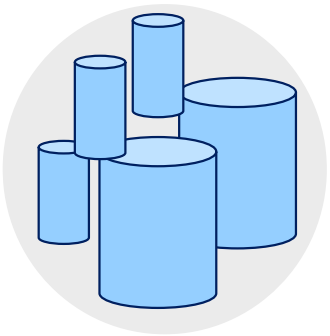


Physical
Datacenter



Mobility and SaaS

Virtualized
Datacenter



BIG Data Solutions

Virtualized
Datacenter
in public cloud

Artificial Intelligence

What is App Modernization?

IT SIMPLIFICATION

- Standardize on the platform and solutions
- Migrate away from legacy systems
- Integrate automation

GROWTH ENABLEMENT

- Focus on the business growth, not the technology
- Easily allow for business scale
- Provide business Insights and Analytics

AGILITY

- Faster deployment to different environments
- Aligned deployment / Less mistakes
- DevOps practices using continuous delivery

REDUCING TCO

- Downsize on infrastructure running cost
- Minimize ongoing maintenance efforts
- Make applications portable

What media should I use to keep backup?

What is the right **size** of **servers** for my business needs?

How do I **deploy** new **code** to my **server**?

What happens in case of **server hardware** failure?

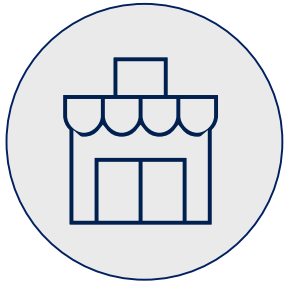
What size of **servers** should I **buy**?

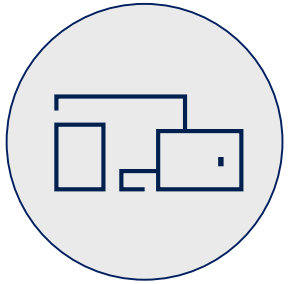
Which packages should be on my **server**?

Who **monitors** my **App**?

How often should I backup my **server**?

How can I **scale** my app?





How can I increase **server** utilization?

Do I need secondary network connection?

Who has **physical** access to my **servers**?

Which OS should I use?

Are my **server** in a secure location?

Who **monitors** my **Servers**?

Do I need a UPS?

What happens if the power goes out?

What storage I need to use?

How many **servers** do I need?

It takes how long to **provision** a new **server**?

How often should I **patch** my **servers**?

How can I dynamically configure my app?

The "evolution" of application platforms

On-Premises

What is the right **size** of **servers** for my business needs?

How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my app?

How often should I **patch** my **servers**?

How often should I backup my **server**?

Which packages should be on my **server**?

How do I **deploy** new **code** to my **server**?

Which OS should I use?

Who **monitors** my App?

The "evolution" of application platforms

On-Premises

IaaS

What is the right **size** of **“servers”** for my business needs?

How can I increase **“server”** utilization?

How many **“servers”** do I need?

How can I **scale** my app?

What are **containers**?

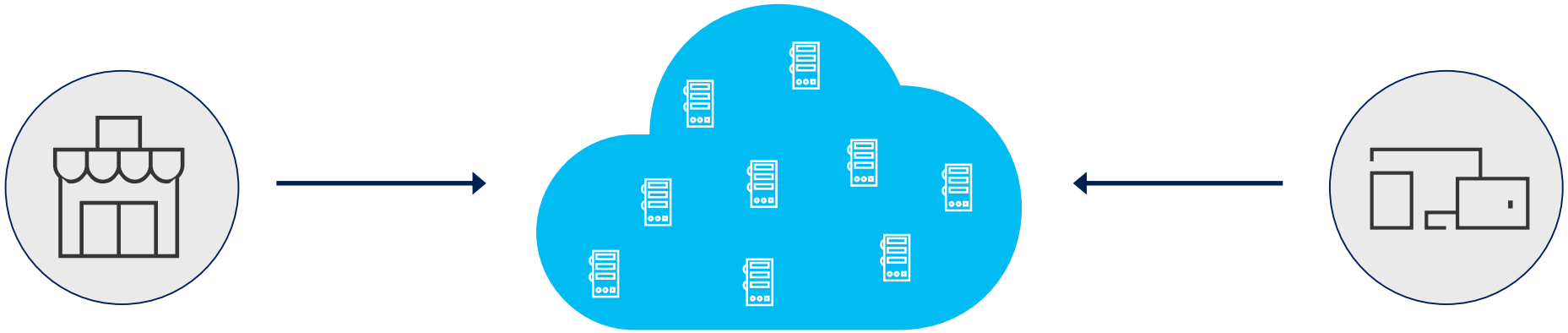
The “evolution” of application platforms

On-Premises

IaaS

PaaS

How do I **(re)architect** my app?

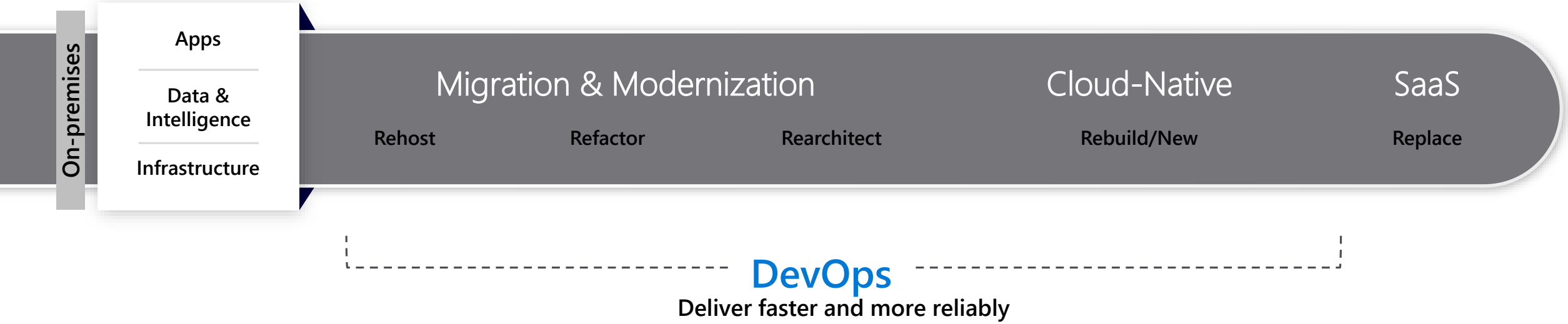


Serverless, the platform for next gen apps

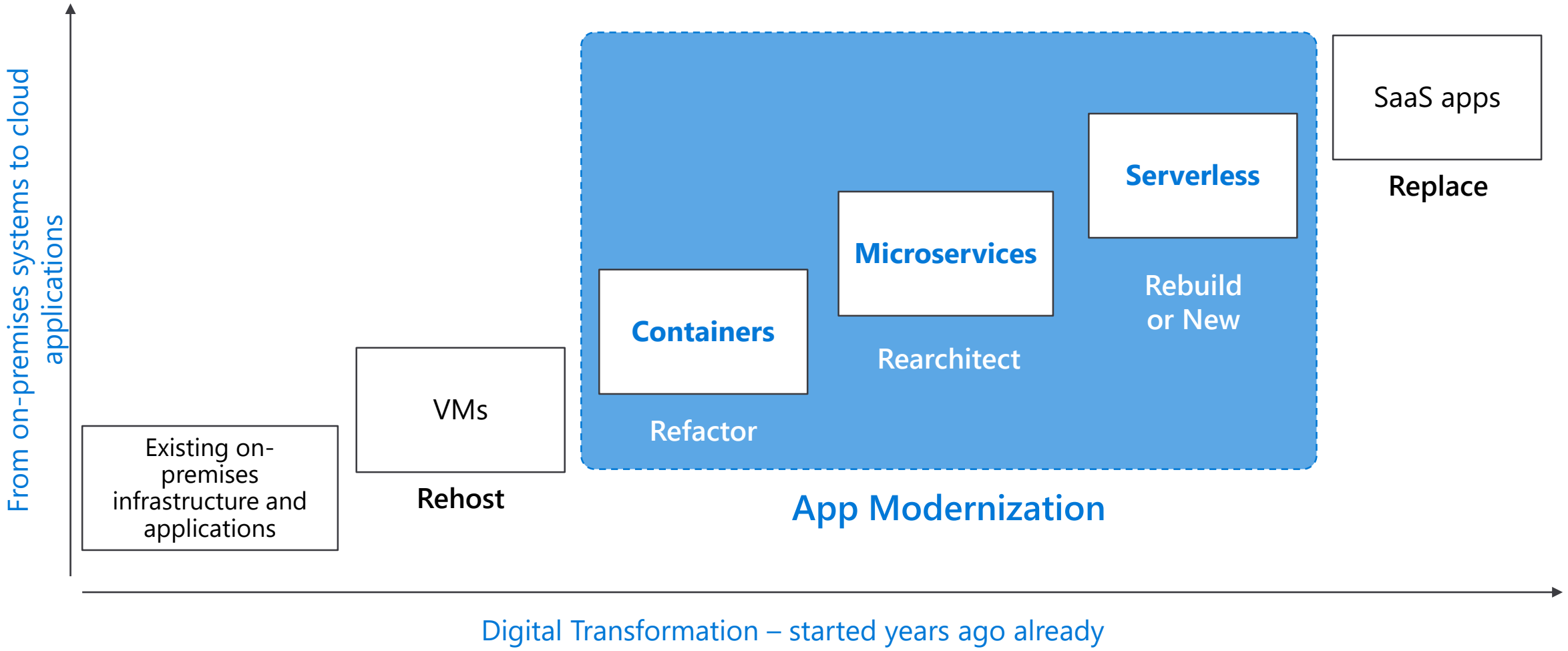
The "evolution" of application platforms

On-Premises → IaaS → PaaS → Serverless

The journey to the cloud

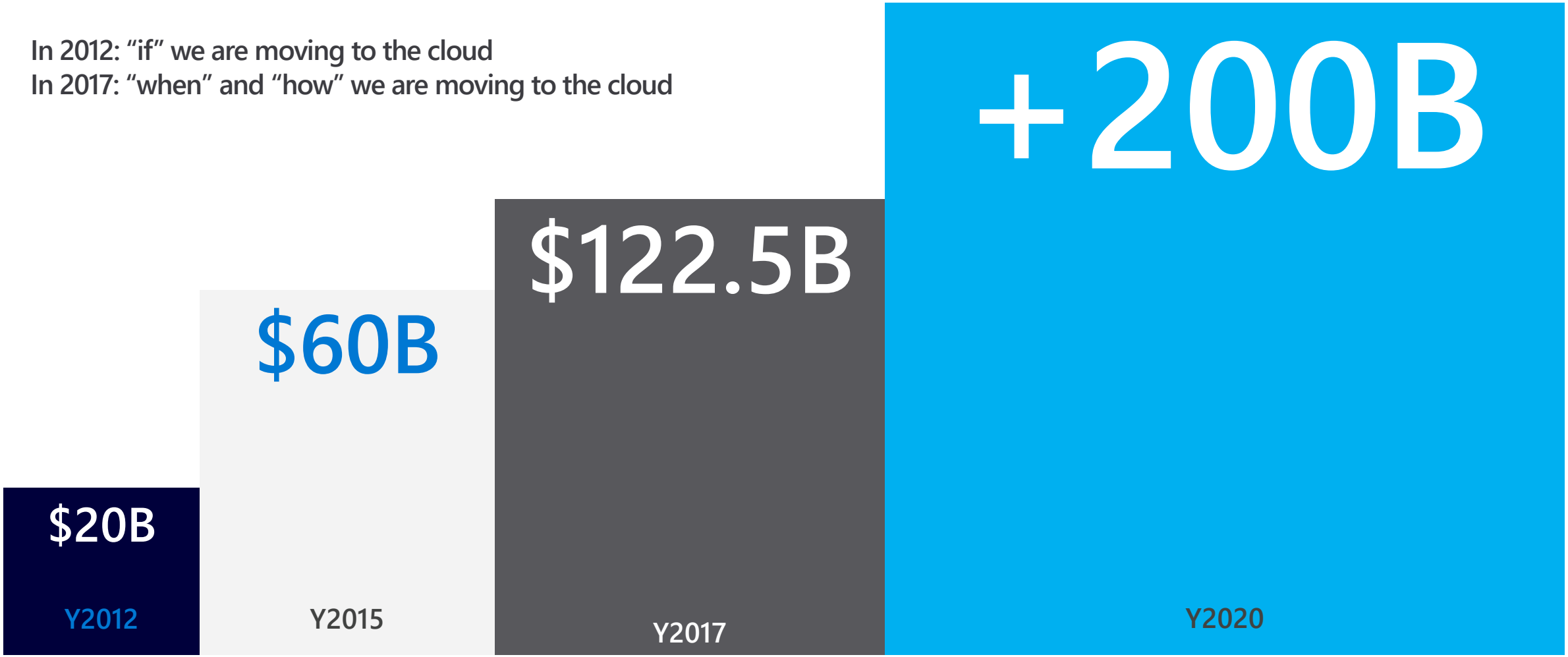


The journey to the cloud

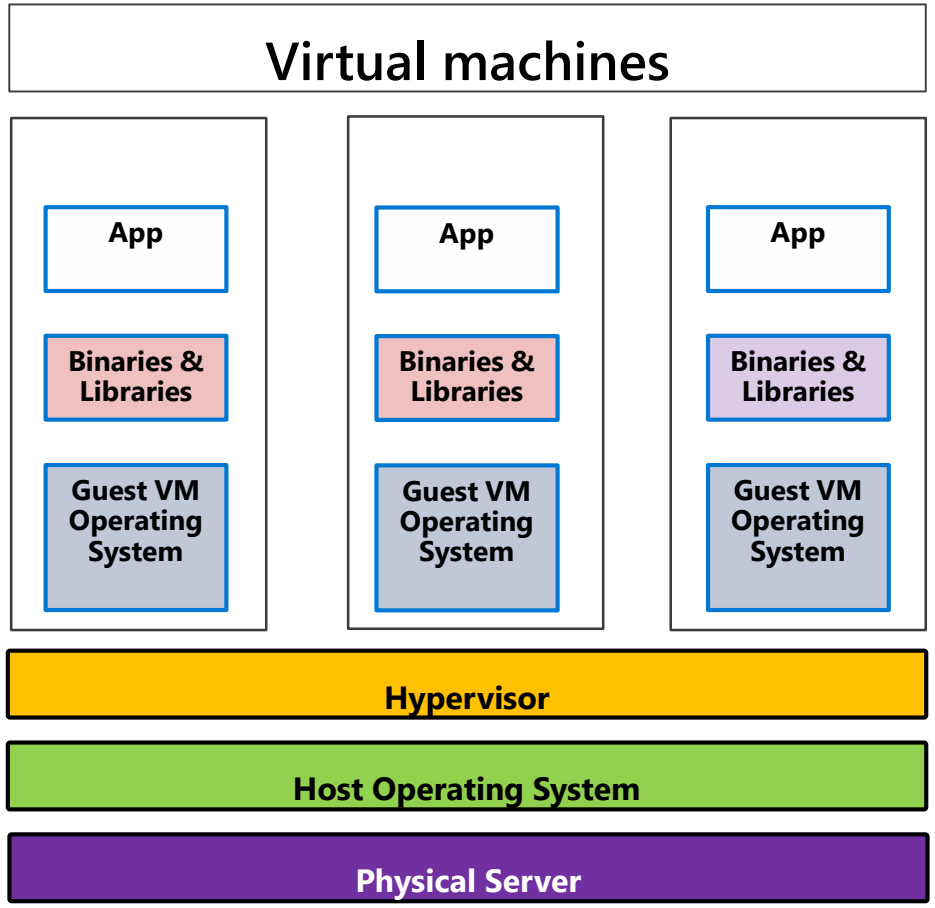


The Cloud has become mainstream

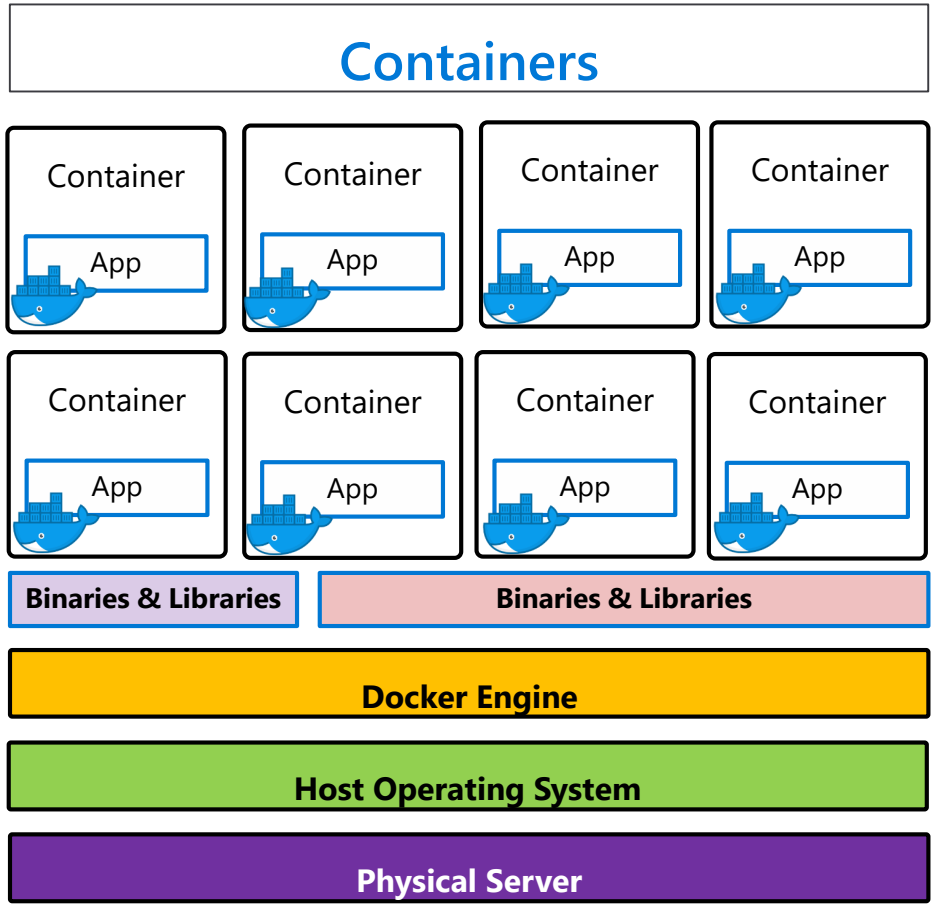
In 2012: “if” we are moving to the cloud
In 2017: “when” and “how” we are moving to the cloud



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)

What are Microservices?

From Monolithic to independent modules

Complex, larger, all-inclusive systems are rearchitected to simpler, smaller, single-purposed modules. Still resulting in a full-fletched application landscape

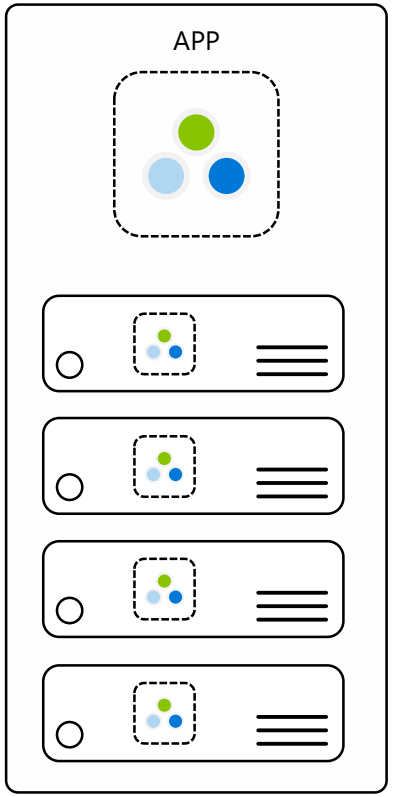
Each module runs a single functionality

These service modules are highly decoupled building blocks that are small enough to implement a single functionality but together can form larger systems

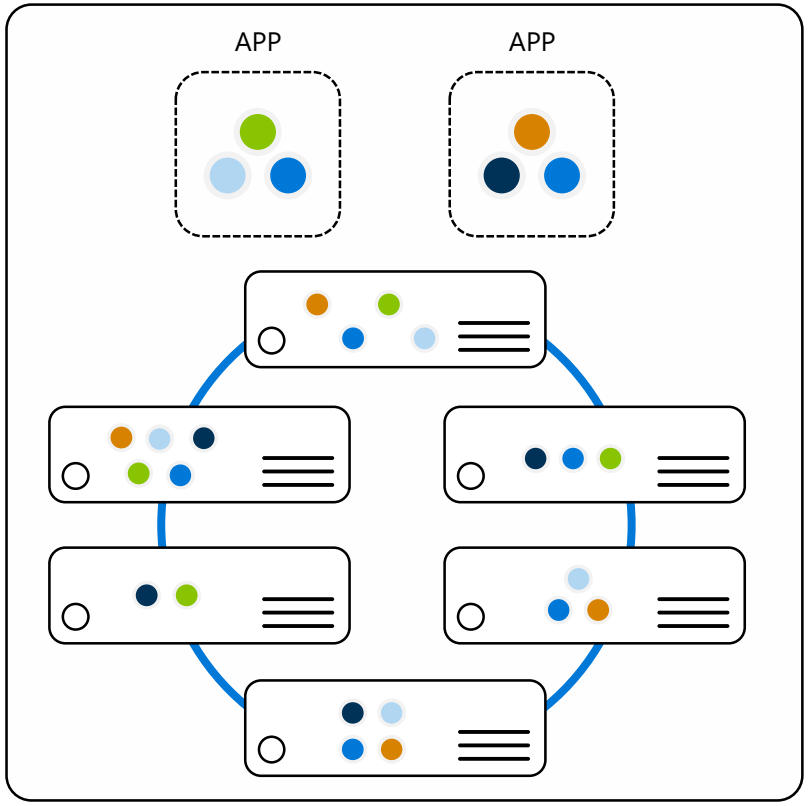
Independently versioned, deployed & scaled

With a microservices architecture, developers can create, manage and improve application services independently, even using different languages

Monolithic
Large, all-inclusive app



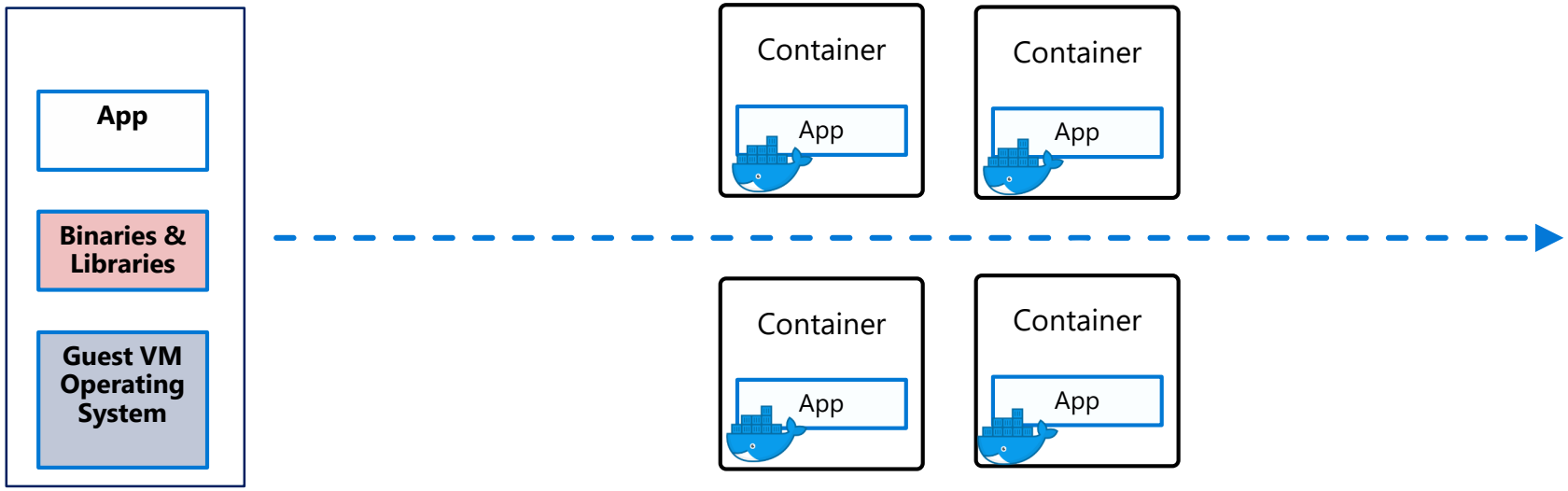
Microservices
Small, independent services



Containers provide the consistent format and isolation desired by microservices.

How do containers help in app modernization?

Containers are stand-alone, smaller silos of app instances, running at scale



Existing/New Applications

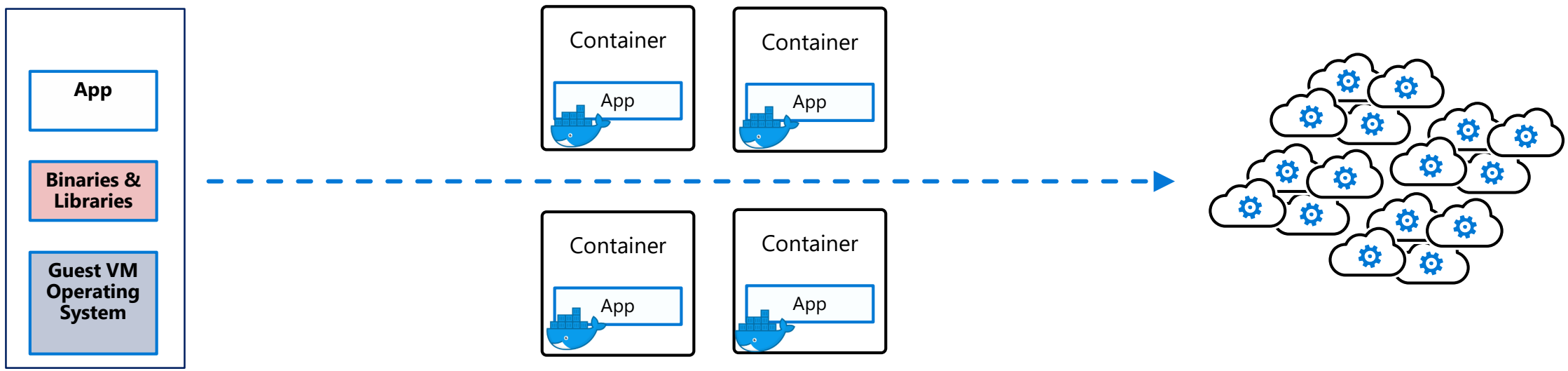
(=rehome)

Lift and shift to containers

(= rehost, refactor)

How do containers help in app modernization?

Containers are not the end-state, but an in-between path to Microservices



Existing/New Applications

(=rehome)

Lift and shift to containers

(= rehost, refactor)

Microservices

(=rearchitect)

Questions Landing Spot

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

© Randy Glasbergen

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

Containers on Azure: covered in this workshop

App Service

Running Docker Containers as Azure Web Apps

Kubernetes Service

Azure Container Services (ACS) with Kubernetes + Azure Kubernetes Service

Container Instance

Running Stand-Alone Docker Containers as a Container Instance

Docker CE

Docker Community Edition on Windows Server 2016/2019

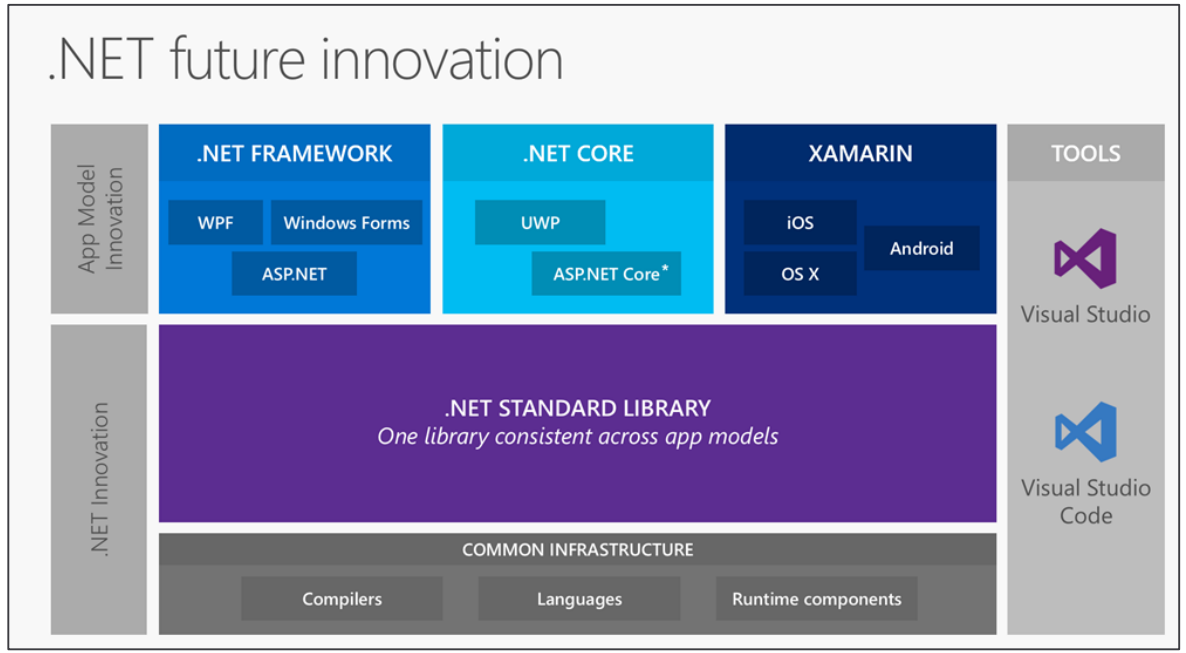
Azure Container Registry

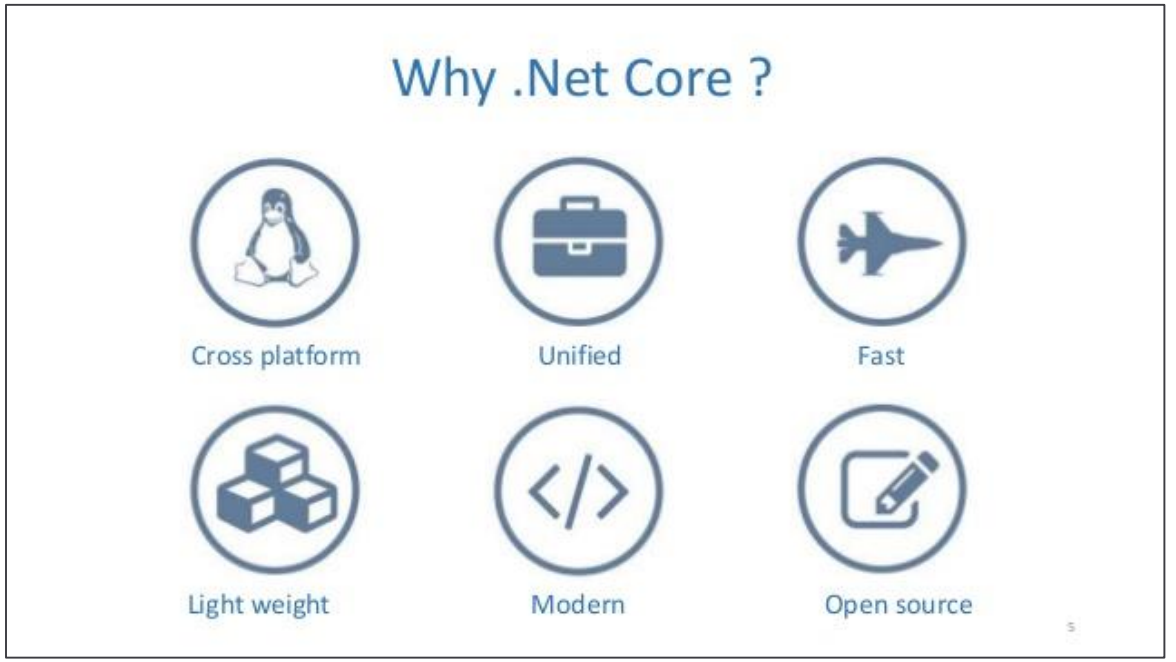
Docker Hub

Choice of developer tools and clients

Isn't a dotnet app the « easy » example?

- .NET is used for 60% of all global Enterprise Web applications
- Containers are not development language specific
- Remember, digital transformation is focusing on the evolution, not the technology or language behind it
- Examples we use in this workshop are just examples





ASP.Net application architecture guides

.NET application architecture guides

Microservices & Docker

Microservices are small, modular, and independently deployable services. Docker containers (for Linux and Windows) simplify deployment and testing by bundling a service and its dependencies into a single unit, which is then run in an isolated environment.

- Architecture e-book: PDF | Web | MOBI | EPUB
- DevOps e-book: PDF | Web | MOBI | EPUB
- Video
- Sample app
- Patterns

Modernizing .NET apps

Lift and shift your existing .NET applications by optimizing your deployments with Windows Containers and by improving your DevOps operations for your dev/test/production environments, ultimately making your application cloud DevOps-ready.

- Architecture e-book: PDF | Web | MOBI | EPUB
- ASP.NET sample apps
- Migrate to cloud
- Video

Azure cloud apps

Production ready cloud applications need to be built for scalability, monitoring, management, security, resiliency, and more. The patterns covered in this guidance include example implementations for Microsoft Azure.

- Quickstart e-book: PDF
- Serverless e-book: PDF | Web | MOBI | EPUB
- Reference architectures
- Best practices
- Patterns
- Free course

ASP.NET web apps

ASP.NET Core allows you to build high-performance, cross-platform web applications. Patterns like MVC and built-in support for Dependency Injection allow you to build applications that are easier to test and maintain.

- Architecture e-book: PDF | Web | MOBI | EPUB
- DevOps e-book: PDF | Web
- Sample app

Xamarin mobile apps

Xamarin allows you to build native Android, iOS, and Windows applications using .NET. Common patterns, such as MVVM, combined with good application layering, will maximize code sharing and result in an application that is easier to understand, test and maintain.

- Architecture e-book: PDF | Web
- Sample app
- Patterns

UWP desktop apps

Windows gives you the tools and capabilities to build modern experiences that empower your customers to do more. The Universal Windows Platform (UWP) lets you create a single app package that can run on a wide range of devices, and the Windows Store provides a unified distribution channel you can use to safely reach customers worldwide.

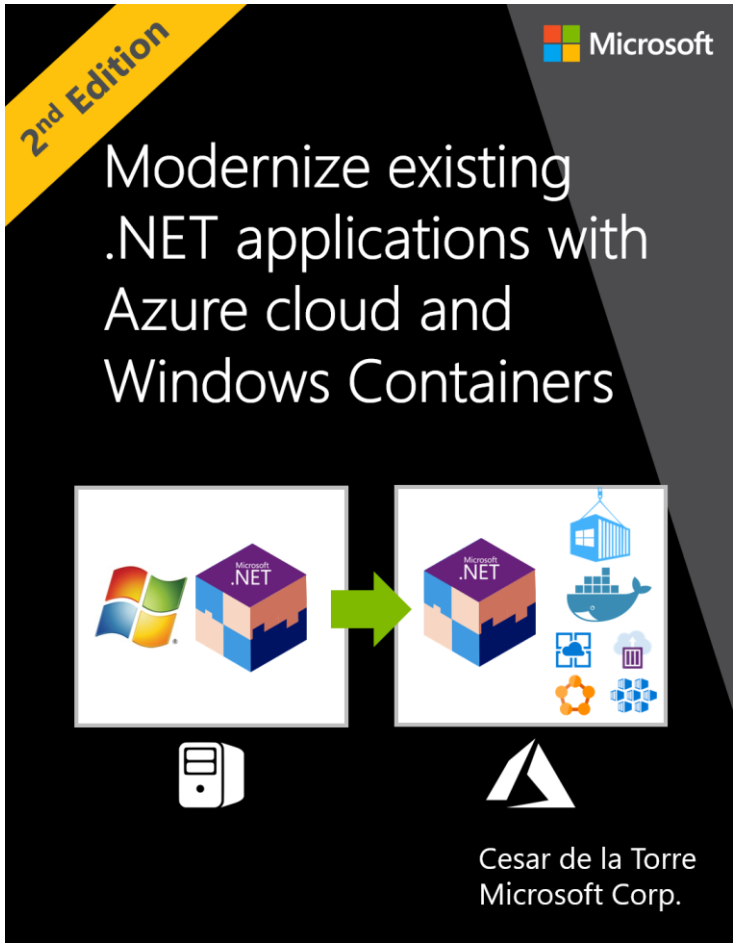
- Sample app

https://www.microsoft.com/net/learn/dotnet/architecture-guides

- Microservices
- Docker
- Windows Containers
- Azure Containers

Guide/eBook and sample apps on Modernize with Windows Containers

eBook/Guide



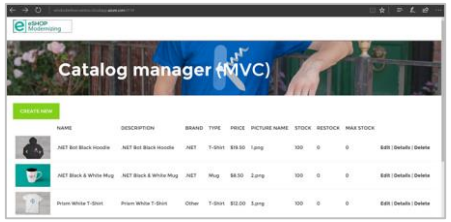
https://aka.ms/liftandshiftwithcontainersebook

Sample Apps for modernization

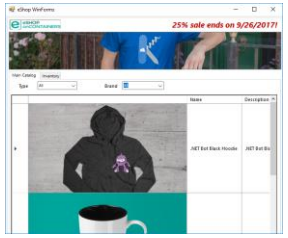
- Intended for .NET developers and solution architects
- Prescriptive guidance on how to modernize your existing .NET apps and migrate to the Azure cloud
- Accompanied with journeys of modernizing existing ASP.NET web apps (**WebForms**, **MVC**) and N-Tier apps (**WCF** + **WinForms**) on eShopModernizing repository.

Client UI apps

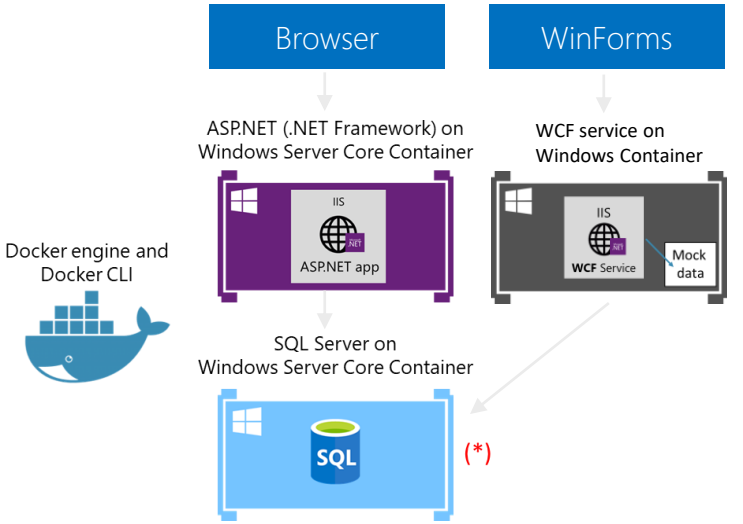
ASP.NET MVC & Web Forms



WinForms



Architecture



https://github.com/dotnet-architecture/eShopModernizing

Microsoft

Maturity model for .Net applications

Existing apps

.NET web apps (on-premises)

Monolithic or **N-Tier** architectures

On-premises

Maturity model for .Net applications

Existing apps

.NET web apps (on-premises)

Monolithic or **N-Tier** architectures

On-premises

Cloud Infrastructure-Ready

Monolithic or **N-Tier** architectures

IaaS

(Infrastructure as a Service)

VMs

Azure

Base Cloud Environment and cross-cutting concerns: Network, Hybrid-cloud, Identity/Auth, Cost control and Operations model

Migrate / Rehost

Maturity model for .Net applications

Existing apps
.NET web apps (on-premises)

Monolithic or **N-Tier** architectures

On-premises

Cloud Infrastructure-Ready

Monolithic or **N-Tier** architectures

IaaS
(Infrastructure as a Service)

Azure

Cloud-Optimized
PaaS

Monolithic or **N-Tier** architectures

Managed services

And
Windows Containers

Azure

Base Cloud Environment and cross-cutting concerns: Network, Hybrid-cloud, Identity/Auth, Cost control and Operations model

Migrate / Rehost

Modernize

Minimal code changes

Maturity model for .Net applications

Existing apps
.NET web apps (on-premises)

Monolithic or **N-Tier** architectures

On-premises

Cloud Infrastructure-Ready

Monolithic or **N-Tier** architectures

IaaS
(Infrastructure as a Service)

Azure

Cloud-Optimized
PaaS

Monolithic or **N-Tier** architectures

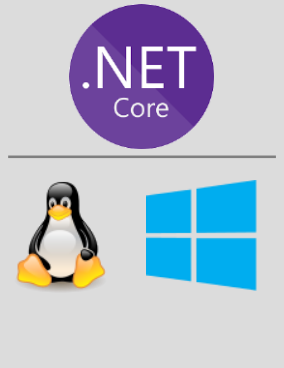
Managed services

Windows Containers

Azure

Cloud-Native
PaaS for microservices and serverless

Microservices and **serverless** architectures



PaaS for containerized microservices
+ Serverless computing
+ Managed services

Azure

Base Cloud Environment and cross-cutting concerns: Network, Hybrid-cloud, Identity/Auth, Cost control and Operations model

Migrate / Rehost

Modernize

Minimal code changes

Architected for the cloud, needs new code

Maturity model for .Net applications

Existing apps
.NET web apps (on-premises)

Monolithic or **N-Tier** architectures

Starting point for this workshop

On-premises

Migration to Web Apps / SQL PaaS

IaaS
(Infrastructure as a Service)

VMs

Azure

Cloud-Optimized
PaaS

Monolithic or **N-Tier** architectures

Managed services

And
Windows Containers

Azure

Cloud-Native
PaaS for microservices and serverless

Microservices and **serverless** architectures

PaaS for containerized microservices
+ Serverless computing
services

Migration to Docker / Container Services

Azure

Base Cloud Environment and cross-cutting concerns: Network, Hybrid-cloud, Identity/Auth, Cost control and Operations model

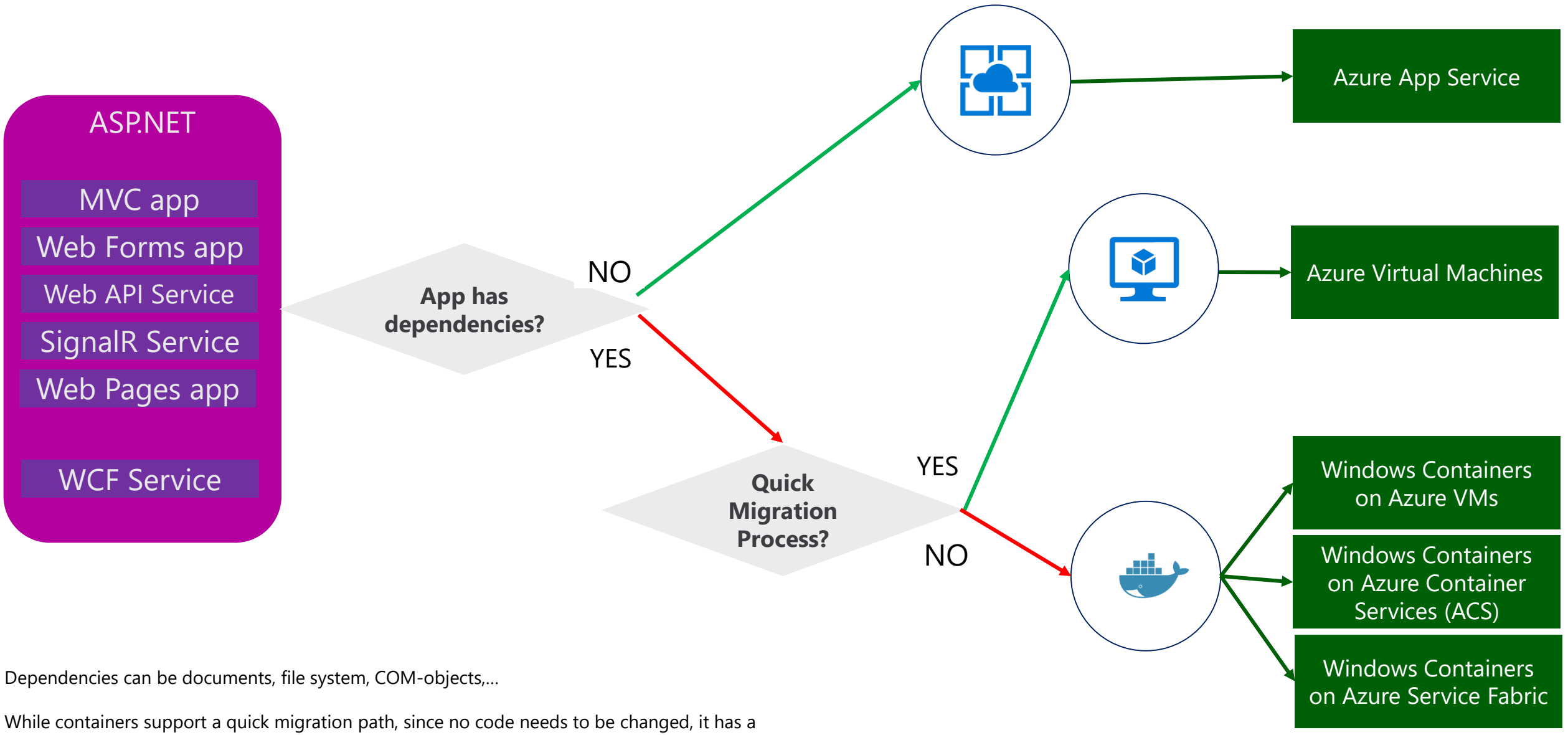
Migrate / Rehost

Modernize

Minimal code changes

Architected for the cloud, needs new code

Azure Compute Decision Tree for .Net Framework app migration



Dependencies can be documents, file system, COM-objects,…

While containers support a quick migration path, since no code needs to be changed, it has a more complex learning curve than migration to Azure VMs

Section Take-Aways

1. Containers are a strategic way to modernize your existing .Net Framework applications
2. Azure provides different choices that enable app modernization
3. Azure natively provides several container services and orchestrators

Questions Landing Spot

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

© Randy Glasbergen

Next Module...

Running Azure Infrastructure and performing VM Lift & Shift migrations

Microsoft

Thank You

© Copyright Microsoft Corporation. All rights reserved.