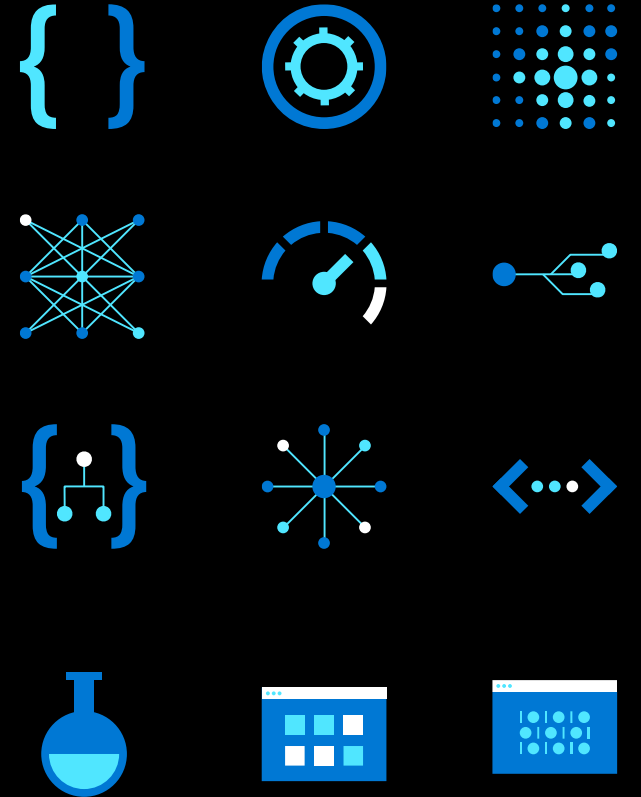


Azure Training Day

Run cloud-native apps with Azure Kubernetes Service



Deploy and run Azure Kubernetes Service clusters

Part 2 of 4 in the Run cloud-native apps with Azure Kubernetes Service series

About us...

John Knightly



Sr. Cloud Solution Architect

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

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 (YOU ARE HERE)**
 - *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

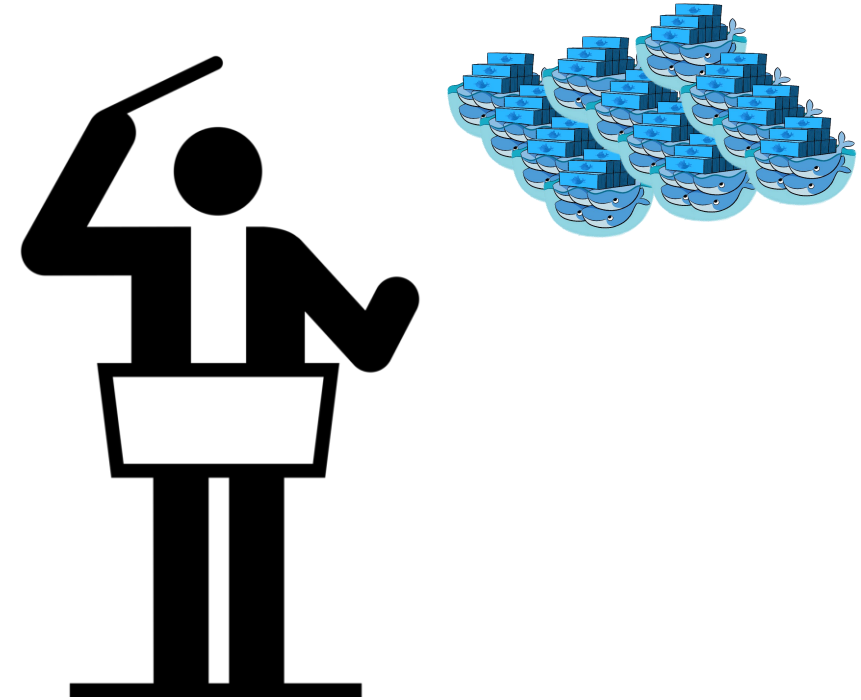
<https://github.com/MSUSDEV/Run-Cloud-Native-Apps-With-AKS>

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

Introduction to Orchestration

Why Container Orchestration?

- Azure Container Instance (ACI) or Azure Webapps for Containers run Containers as « **stand-alone** » resources
- **Orchestration gives you:**
 - Auto-Scaling
 - Rolling Upgrades
 - Service Discovery
 - Integrated Load-Balancing
 - Affinity / anti-affinity
 - Scaling Options
 - Custom Resource Sizing
 - ...

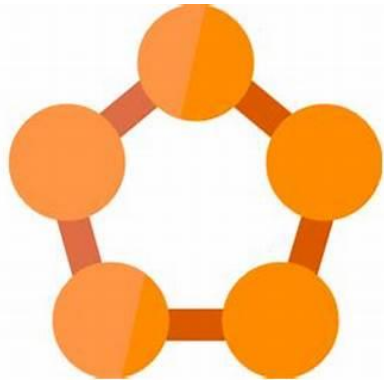


What Container Orchestration Is There?

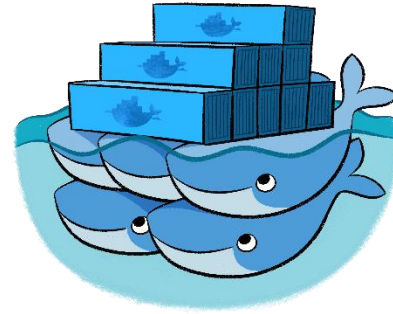
What solutions are available?



Kubernetes



Service Fabric



Docker Swarm



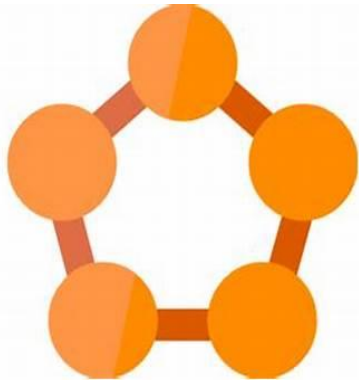
DC/OS Mesosphere

What Container Orchestration Is There – in Azure?

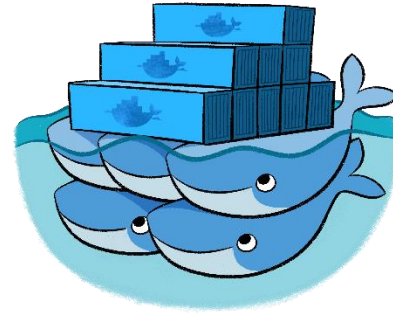
What Orchestration tools are available in the Azure Platform?



**Azure
Kubernetes
Service (AKS)**



**Azure
Service
Fabric**



**Docker
Enterprise
(IAAS)**



**DC/OS Mesosphere
(IAAS)**

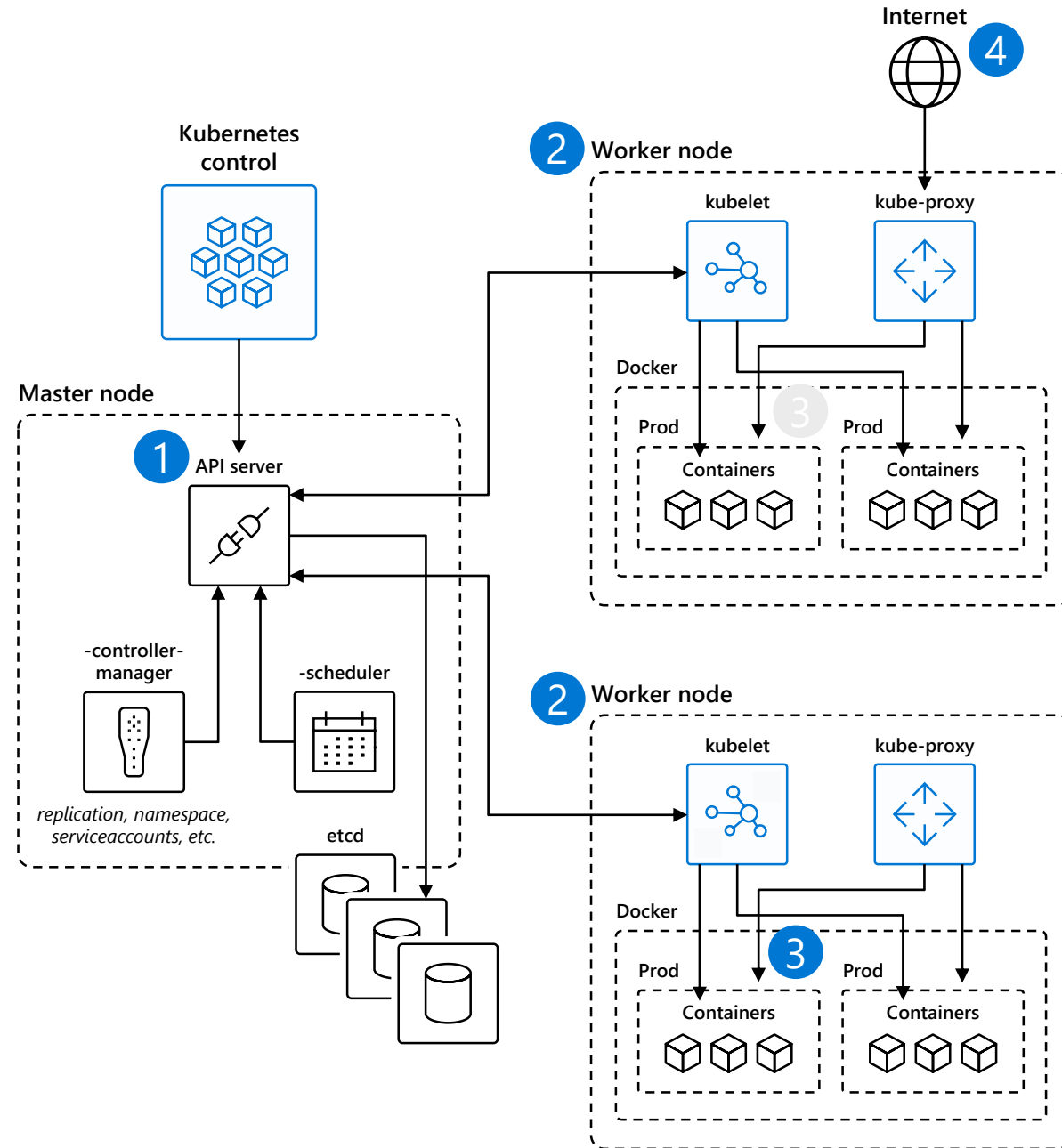
What makes Kubernetes the “Go-To” orchestrator?

The preferred Azure Container Orchestrator

- **Portable across clouds**
 - Public, Private, Hybrid, Multi-Cloud
- **Extensible**
 - Modular, pluggable, composable
- **Self-Healing**
 - Auto-placement, Auto-Restart, Scaling,...

Base Kubernetes Architecture

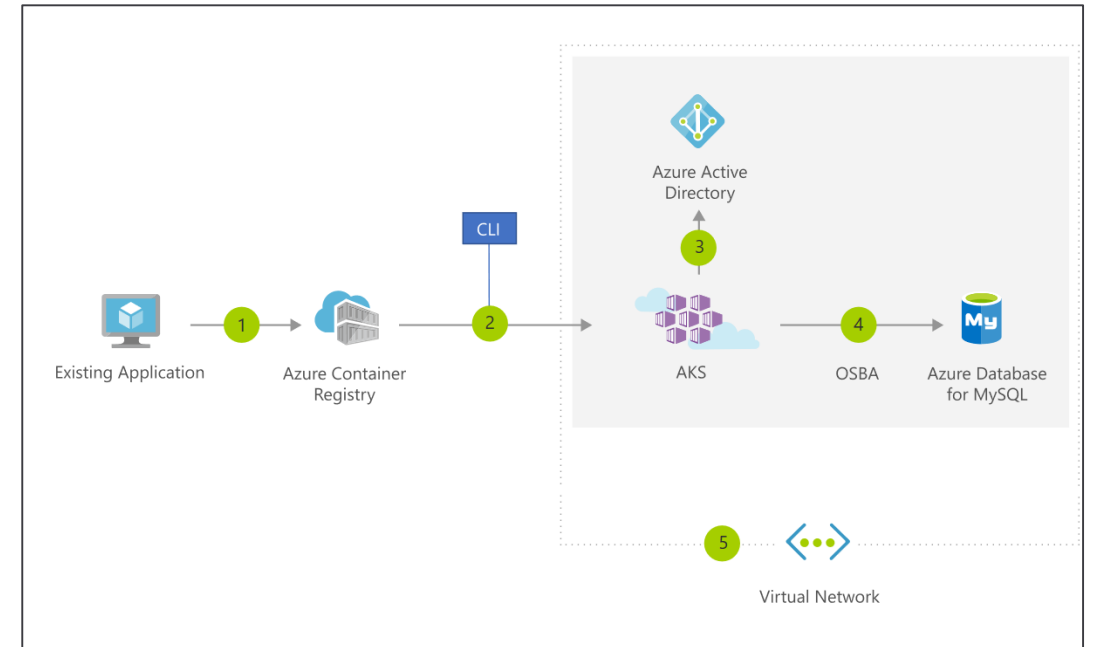
1. **Kubernetes users communicate with API server and apply desired state**
2. **Master nodes actively enforce desired state on worker nodes**
3. **Worker nodes support communication between containers**
4. **Worker nodes support communication from the Internet**



Azure Kubernetes Services (AKS) in-depth

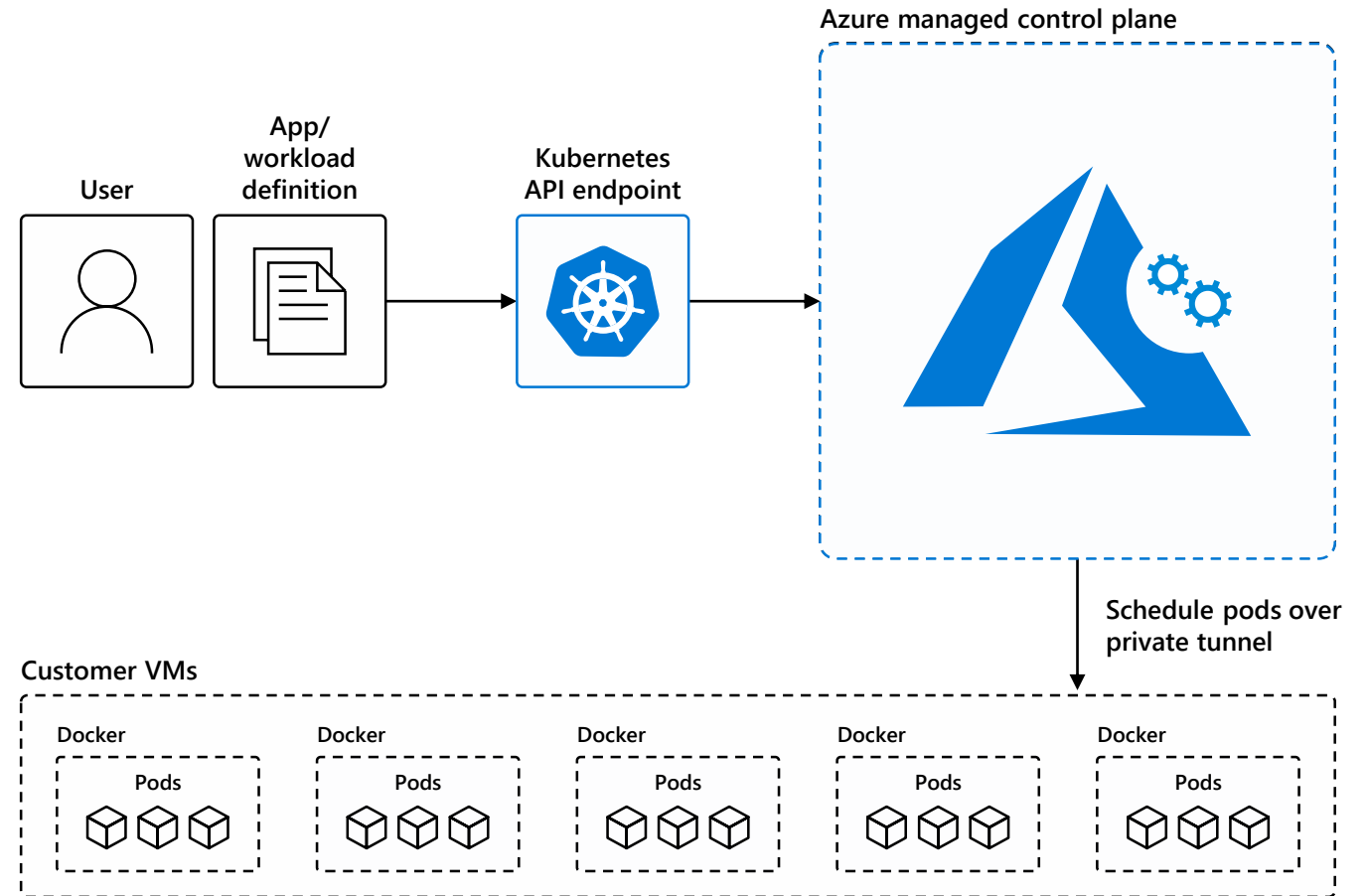
The preferred Azure Container Orchestrator

- Easily deploy from Portal or CLI
- IAC with ARM or Terraform
- Start with a single node and scale out
- Deploy apps into AKS using Azure DevOps CI/CD Pipeline
- **NOW supports both** Linux and Windows as Nodes

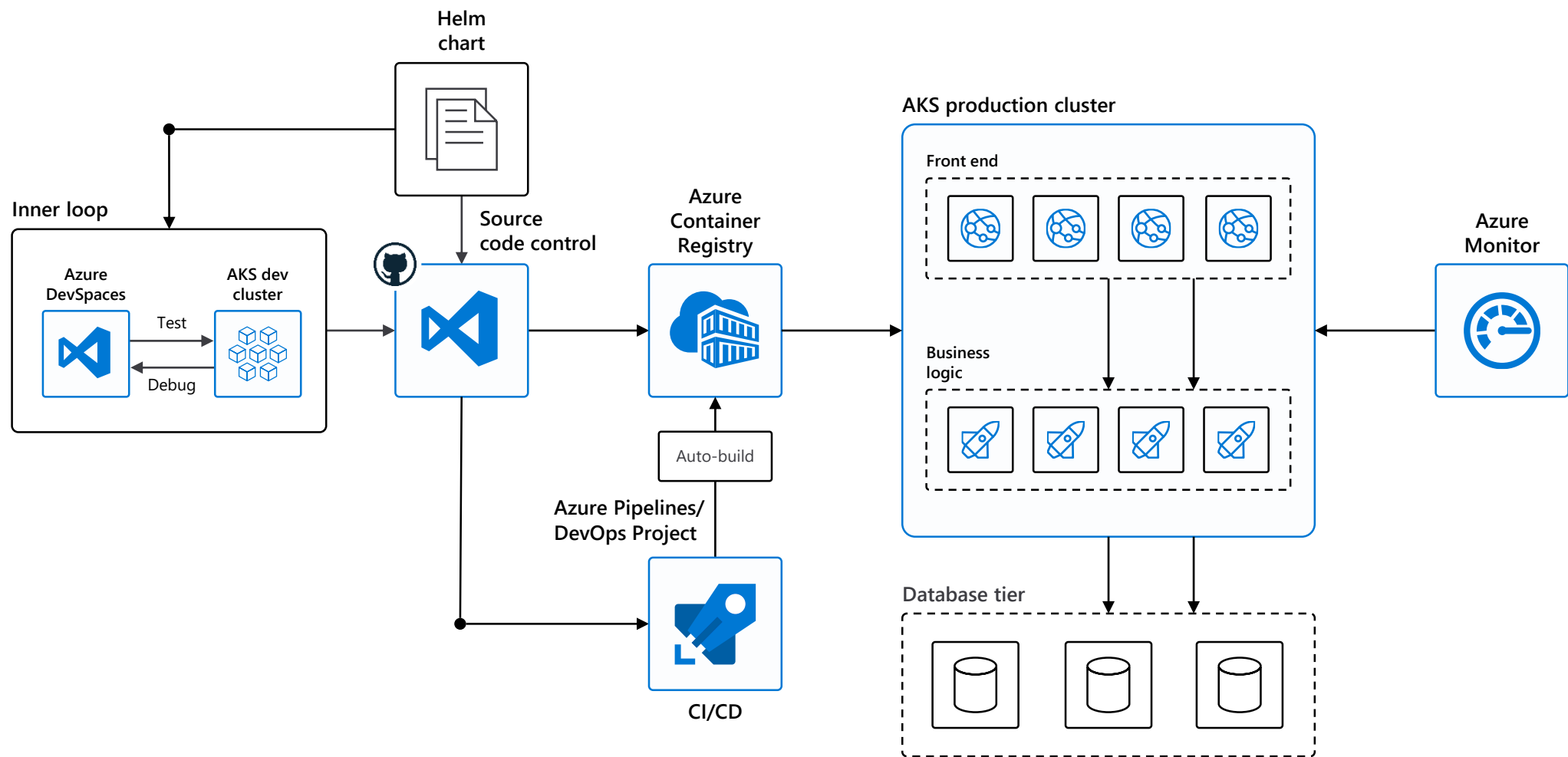


How managed Kubernetes on Azure works

- Automated upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge (you still pay for the worker nodes)



AKS end-to-end experience inside Azure DevOps



Developer owns this

ACR and Azure DevOps

Azure AKS Control Plane

Azure Monitor

Deploying Azure Kubernetes Services

Using Azure CLI

```
az aks create --resource-group AKSnativeRG --name AKSCluster --node-count 1 --enable-addons monitoring --generate-ssh-keys
```

```
az aks get-credentials --resource-group AKSnativeRG --name AKSCluster
```

Managing Azure Container Services

KubeCTL Command Line

az aks install-cli

Kubectl get nodes

```
Administrator: Windows PowerShell
PS C:\Users\labadmin> kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
aks-nodepool1-20062427-0            Ready     agent    22h    v1.9.9
PS C:\Users\labadmin>
```

Demo

Deploying Azure Kubernetes Services using Azure CLI

Integrating AKS with ACR

Running a Container Registry Image inside Azure Container Services

Kubectl create -f “path to YAML-file”

```
Administrator: Windows PowerShell
PS C:\DockerImage1> kubectl create -f .\Kubernetes2.yml
deployment.apps "akshelloworld" created
service "akshelloworld" created
```

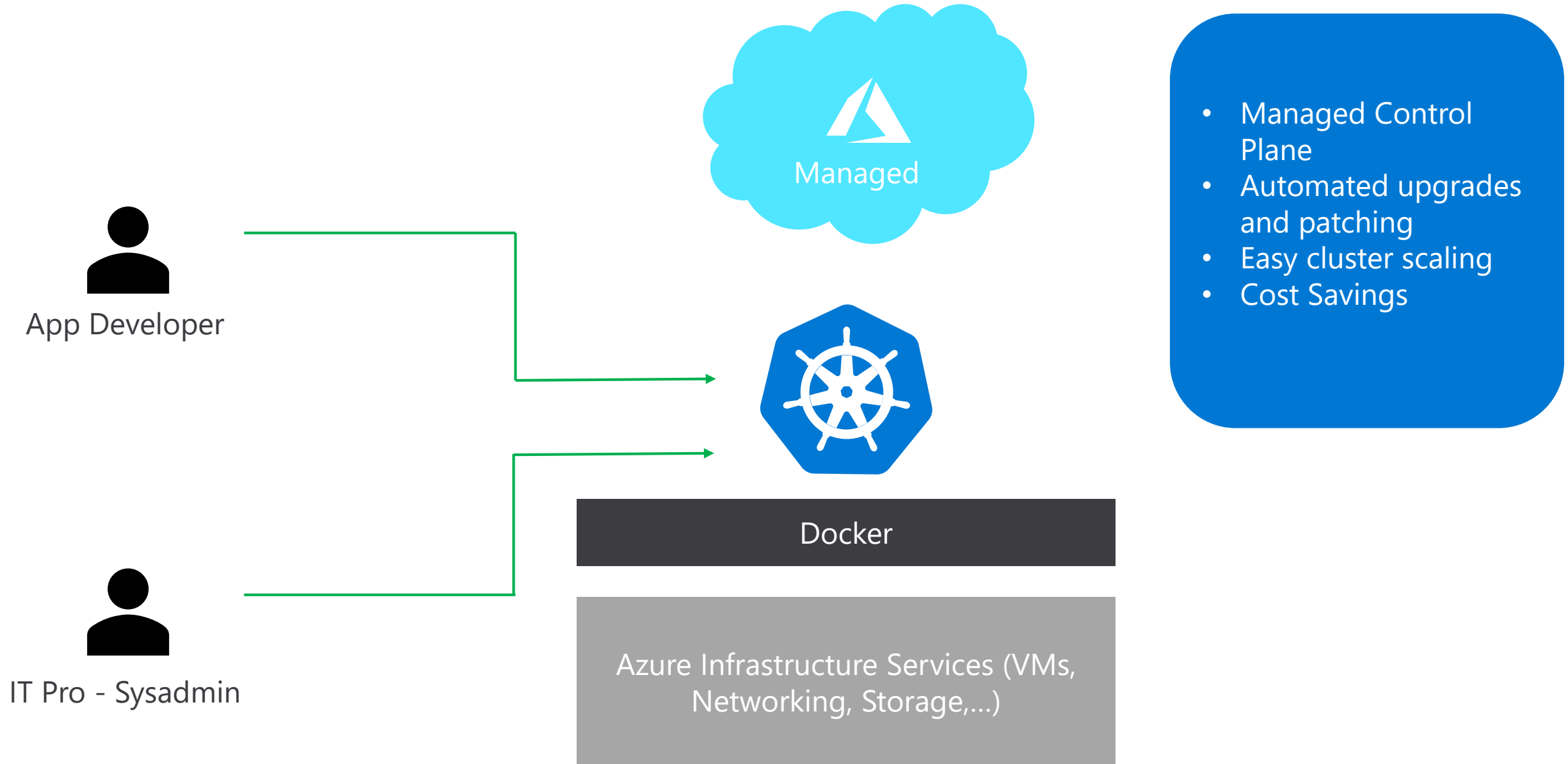
```
Administrator: Windows PowerShell
PS C:\DockerImage1> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
adsakssample-6d7c8cf5cd-9brgr       0/1     ImagePullBackoff    0           42m
akshelloworld-64dbbb7cf8-vfqnc      1/1     Running         0           12m
dockerwebvmsample-79947845f6-jwr/p  0/1     ImagePullBackoff    0           42m
dockerwebvmsample2-77cd55c9bd-kkcfh 0/1     ImagePullBackoff    0           42m
newadsakssample-6486f76985-p4r42    0/1     ImagePullBackoff    0           41m
newdockerwebvmsample-54dff974d-qpvr4 0/1     ImagePullBackoff    0           42m
PS C:\DockerImage1>
```

```
app: akshelloworld
spec:
  containers:
  - name: adsacr
    image: docker.io/microsoft/aci-
helloworld
    ports:
    - containerPort: 80
  imagePullSecrets:
  - name: adsacr-auth
```

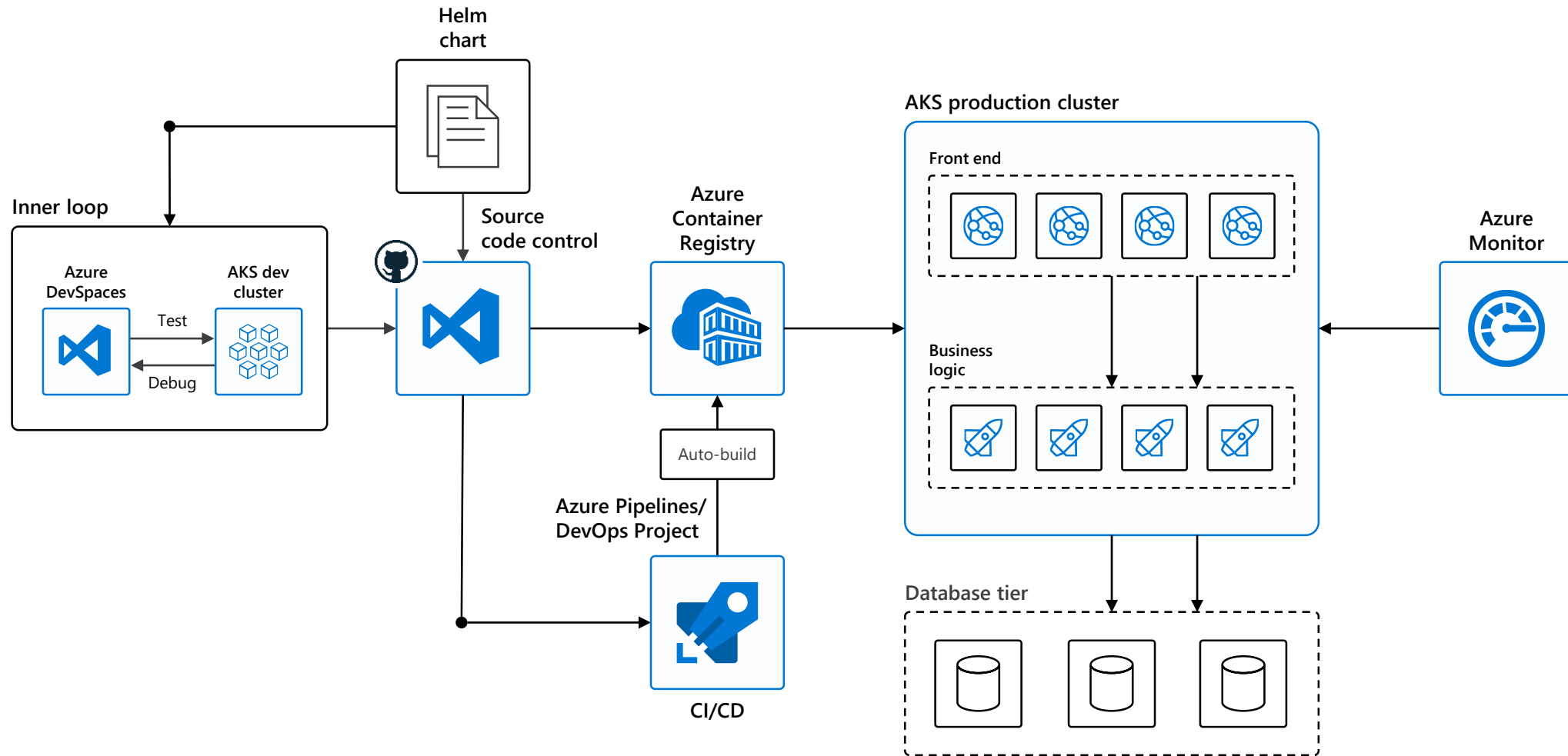
Demo

Running a Docker Hub image in Azure Kubernetes Services

Azure Kubernetes Service Architecture



AKS end-to-end experience inside Azure DevOps



Developer owns this

ACR and Azure DevOps

Azure AKS Control Plane

Azure Monitor

Demo

AKS end-to-end integration with Azure DevOps

Azure Dev Spaces

Azure Dev Spaces helps development teams be more productive on Kubernetes in the following ways:

- Minimize local dev machine setup for each team member and work directly in AKS, a managed Kubernetes cluster in Azure.
- Rapidly iterate and debug code directly in Kubernetes using Visual Studio 2017 or VS Code.
- Generate Docker and Kubernetes configuration-as-code assets for you to use from development through to production.
- Share a managed Kubernetes cluster with your team and collaboratively work together. Develop your code in isolation, and do end-to-end testing with other components without replicating or mocking up dependencies.

Deploying Azure Dev Spaces in AKS

Using Azure CLI

```
az aks use-dev-spaces --resource-group AKSnativeRG --name AKSCluster
```

=> Your AKS Cluster must already be up and running 😊

Demo

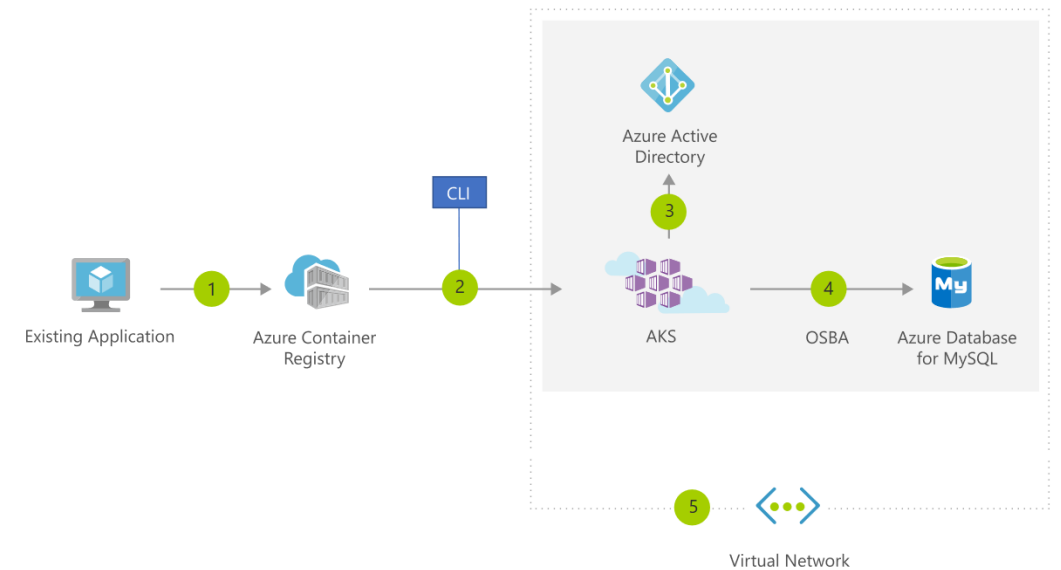
Azure Dev Spaces with AKS

Deploying AKS for Windows

AKS for Windows Nodes

The preferred Azure Container Orchestrator for Windows Containers

- Provides AKS Master (Linux)
- Provides AKS Nodes (Win2019)
- It's AKS as you know AKS
- Deployment is done using AKS-Engine
- Management is done using Kubectl (No Azure Portal integration yet)



Demo

AKS-Engine for Windows cluster deployment walkthrough

Section Take-Aways

1. Azure offers several Container Orchestration Solutions
2. Kubernetes – and thus Azure Kubernetes Services (AKS) is the « go-to » orchestration service
3. AKS supports both Linux and Windows (2019) Nodes, allowing to run about any containerized application, no matter the platform characteristics

Questions Landing Spot

“...If you want good answers,
ask better questions...”

© Randy Glasbergen



Azure

Next Module...

Azure DevOps Pipelines



Appendix

Lab dependencies and alternate path

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)

Create container instance

✓ Validation passed

Basics Networking Advanced Tags Review + create

Basics

Subscription	007FFFLearning Labs
Resource group	simplwebappRG
Region	(Europe) North Europe
Container name	nodbcontainer
Image type	Public
Image name	simplcommerce/ci-build
OS type	Linux
Memory (GiB)	1.5
Number of CPU cores	1
GPU type	None
Number of GPU cores	0

Networking

Include public IP address	Yes
Ports	80 (TCP)

Advanced

Restart policy	On failure
Command override	[]

Tags

(none)

Full workshop

Dependencies from first series

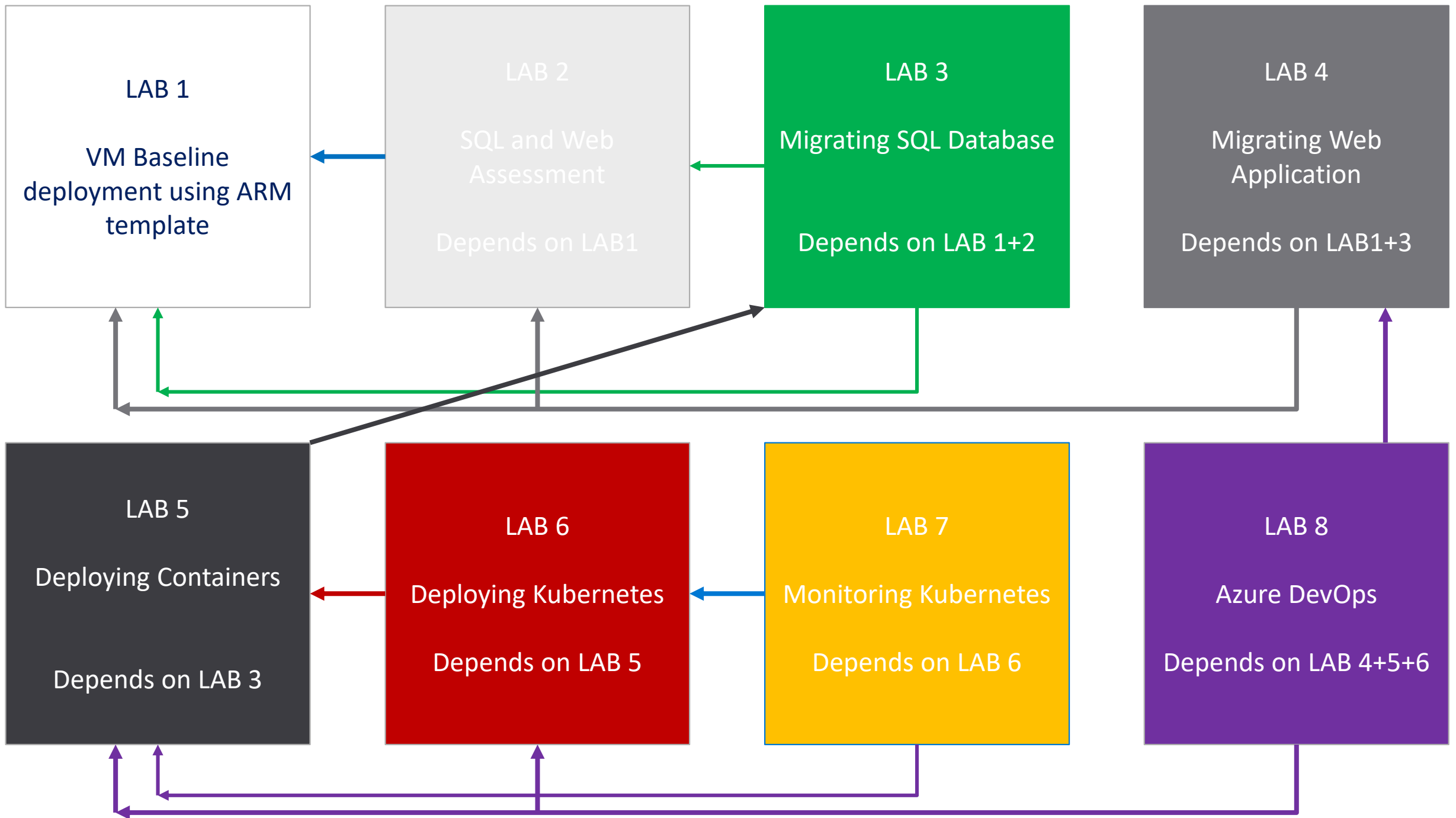
The lab guide assumes that the learner has completed the first 3 labs from the previous series “[Migrate a web app to Azure](#)” and the first lab of this series

[Deploying VM baseline using ARM Templates](#)

[Performing Assessments](#)

[Migrating SQL Databases](#)

[Deploying and running Azure containers](#)





Thank You