Microsoft

Azure

Azure Developer Series

Application Migration to Azure

Peter De Tender

@pdtit

@007FFFLearning

CEO & Lead Technical Trainer at
007FFFLearning.com

April 2019

About Me…

Peter De Tender – MCT, Azure MVP

- CEO and Lead Technical Trainer of 007FFFLearning.com, +20 years IT experience, mainly datacenters and Microsoft Infrastructure background
- Full-time in Azure since 2013 (Readiness & Architect)
- Azure Advisor, Azure Certified Architect
- Technical Writer, Book author, Courseware Creator
- Living in Belgium, but traveling worldwide 90% of my time, helping larger Microsoft Partners, customers and Microsoft FTEs in learning about and using Azure, by providing workshops with passion

peter@pdtit.be

@pdtit @007FFFLearning

http://www.facebook.com/pdtit

http://www.linkedin.com/in/pdtit

Setting the scene

Overview of the workshop

About the workshop content…

About:
In this workshop, you will learn how to build a proof of concept (POC) that will transform an existing ASP.NET-based Web application to a container-based application. This POC will deliver a multi-tiered web app solution from a Virtual Machine architecture into Azure, leveraging Azure WebApps and different Azure container solutions available today. You will also migrate the underlying database from a SQL 2014 Virtual Machine architecture to SQL Azure. Easter Bonus: Every now and then, we will showcase similar steps using a Node.JS and MongoDB, migrating to Azure Web Apps, Containers and CosmosDB.

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 SQL Azure PaaS solutioning.

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:
16 hours (+/- 10 hours presentations, 6 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: Infrastructure as Code using ARM templates
- Module 3: Azure Database Solutions – SQL Azure
- Module 4: Azure App Services – Azure Web Apps (.NET + Node.JS)
- Module 5: Introduction to Docker
- Module 6: Deploying Azure Container Registry / Azure Container Instance
- Module 7: Migrating Apps to Azure Container Services / Kubernetes Services
- Module 8: ACS / AKS Management and Monitoring

Workshop Agenda – Hands-On-Labs

Learn by doing…

- Module 2: Infrastructure as Code using ARM templates
  - **Lab 1:** Setup your Azure subscription and deploy the source Virtual Machine environment with Visual Studio 2017
- Module 3: Azure Database Solutions – SQL Azure
  - **Lab 2:** Migrating a SQL VM database to SQL Azure using SQL Management Studio
- Module 4: Azure App Services – Azure Web Apps
  - **Lab 3:** Migrating your legacy ASP.NET application to Azure Web Apps with Visual Studio 2017
  - Easter Egg Bonus: Deploying a Node.JS app with MongoDB / CosmosDB
- Module 5: Introduction to Docker
  - **Lab 4:** Containerizing your legacy ASP.NET application with Docker CE for Windows

Workshop Agenda – Hands-On-Labs

Learn by doing…

- Module 6: Deploying Azure Container Registry / Azure Container Instance
  - **Lab 5:** Using Azure Container Registry, Azure Container Instance
- Module 7: Migrating Apps to Azure Container Services / Kubernetes Services
  - **Lab 6: Deploying Azure Container Services with Kubernetes and running Pods**
  - **Lab 7: Deploying Azure Kubernetes Services**
- Module 8: ACS / AKS Monitoring and Operations
  - **Lab 8:** Integrating ACS monitoring with Azure Monitor and Deploying Kubernetes Dashboard

Node.JS and Cosmos DB labs are available on request

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,…)
- Lab consumption estimate: $15-35 (when shutdown all resources)

Questions and HOL support

msdevseriessupport@007FFFLearning.com

Subject: Azure Developer Series – Containers

Response Time: within 4-8 hours

Check GitHub for FAQ and Updates:
http://www.github.com/007FFFLearning/MSDevSeriesSupport

Microsoft

Azure

Application Migration

Container Orchestration with ACS and AKS

Peter De Tender

@pdtit @007FFFlearning

Key Objectives

What you will learn in this section

- Azure Container Services (ACS)
- Azure Kubernetes Services (AKS)
- Azure Dev Spaces

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?

What tools are available?

Kubernetes

DC/OS Mesosphere

Docker Swarm

Service Fabric

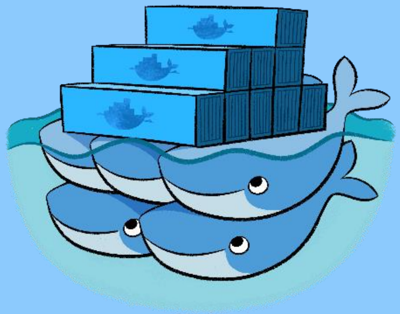
What Container Orchestration in Azure?

Azure Container Services (ACS)

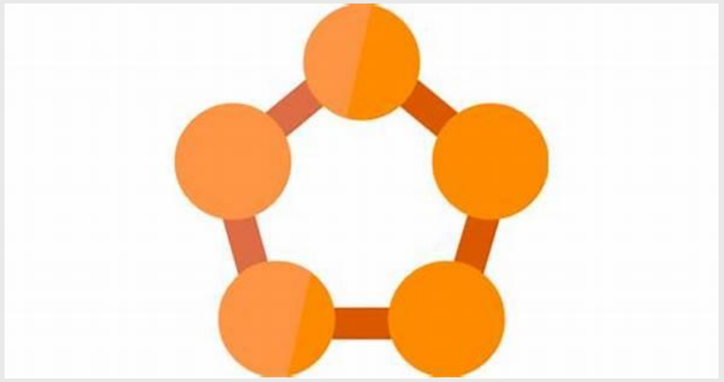
ACS will be deprecated by January 30, 2020

Kubernetes

DC/OS Mesosphere



Docker Swarm



Service Fabric

What Container Orchestration in Azure?

Azure Kubernetes Services (AKS)

Kubernetes

DC/OS Mesosphere

~~Docker Swarm~~

Docker Enterprise

Service Fabric

What makes Kubernetes the standard 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,…

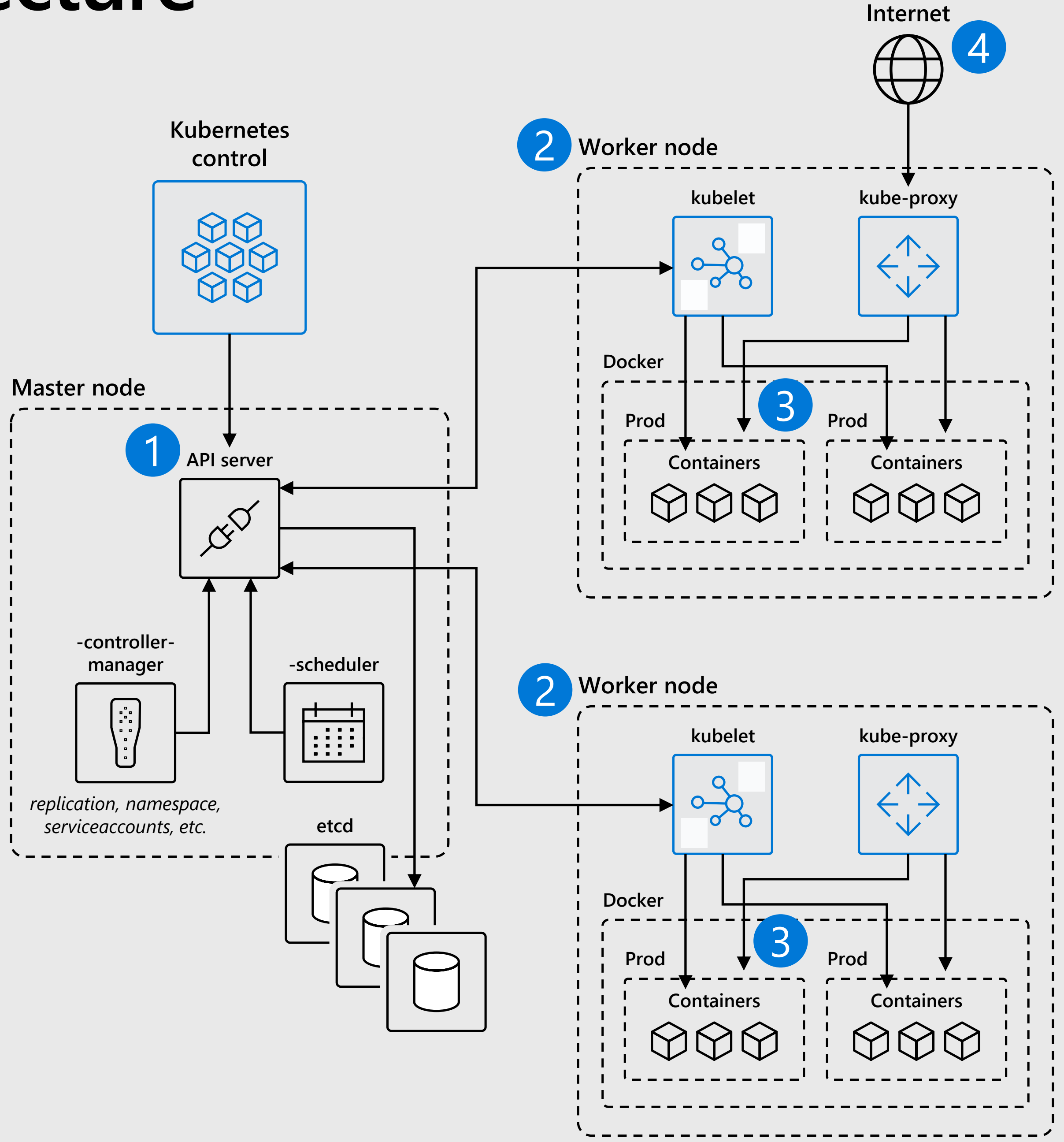
Base Kubernetes Architecture

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

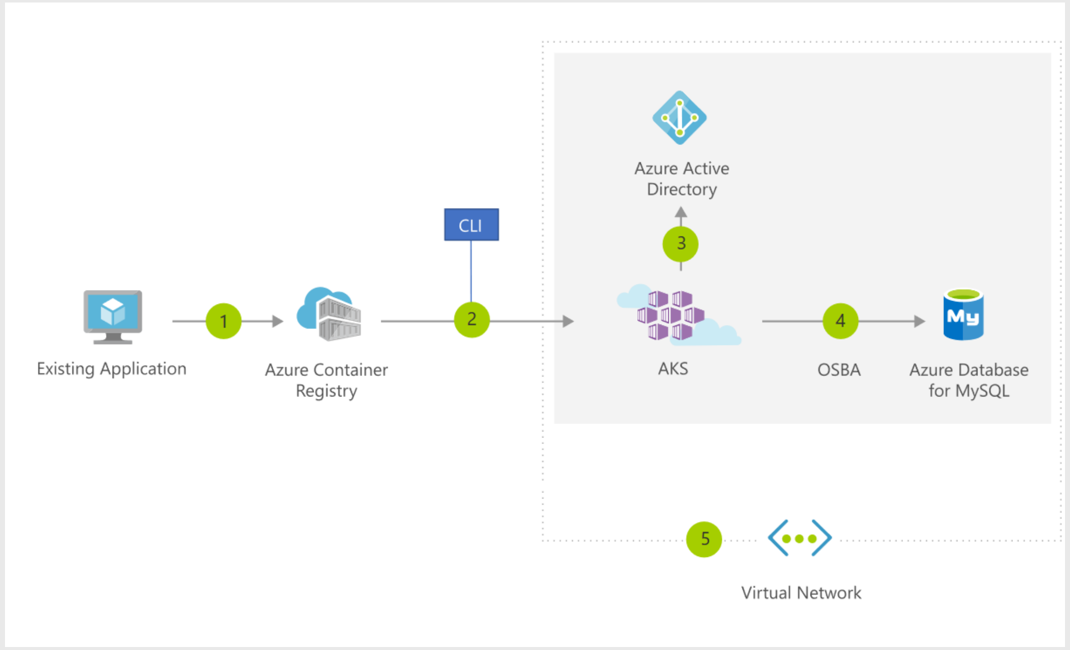
4. Worker nodes support communication from the Internet



Azure Kubernetes Services (AKS) in-depth

The preferred Azure Container Orchestrator

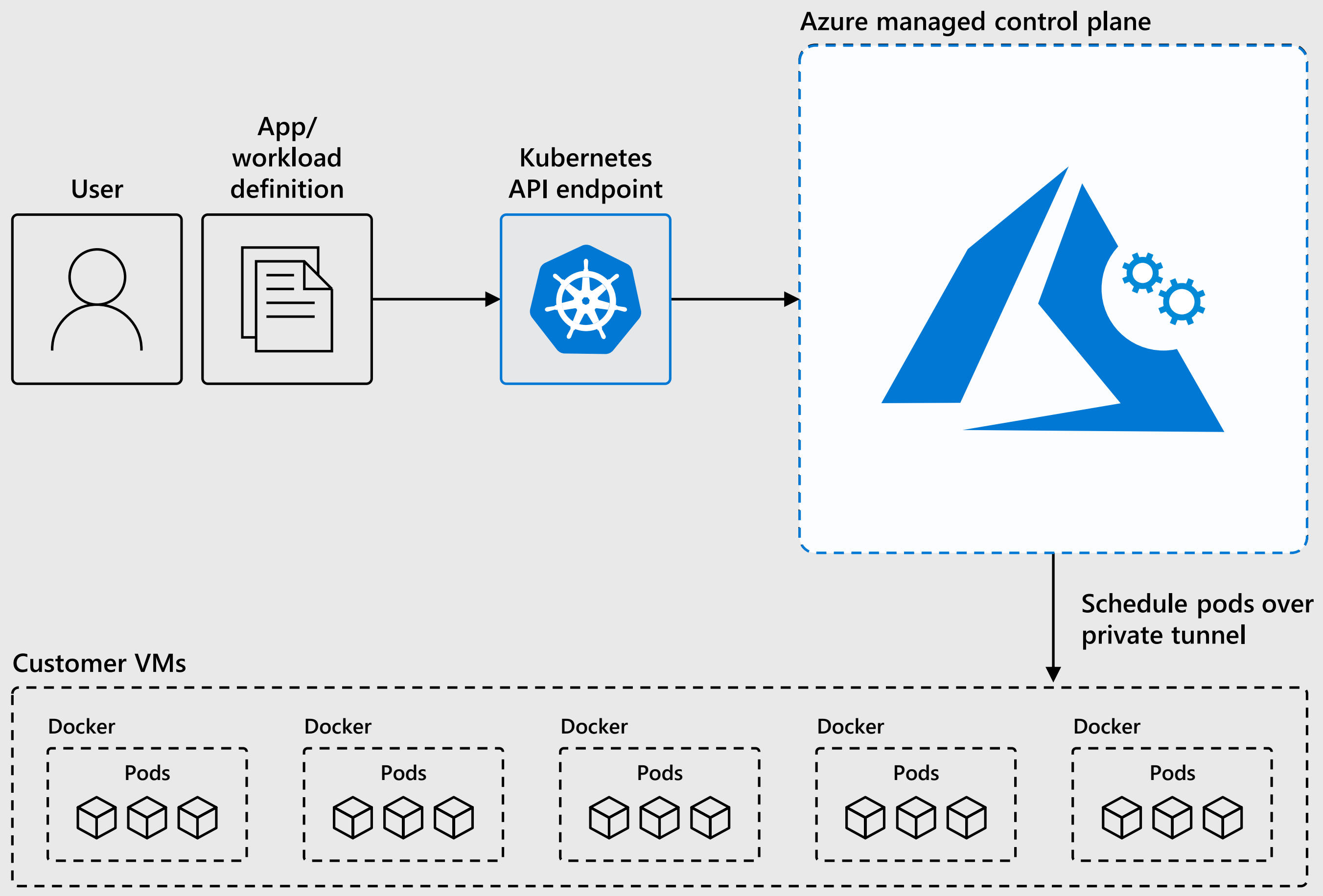
- Easily deploy from Portal or CLI
- IAC with ARM or Terraform
- Start with a single node and scale out
- Deploy apps into AKS using Azure DevOps CI/CD Pipeline



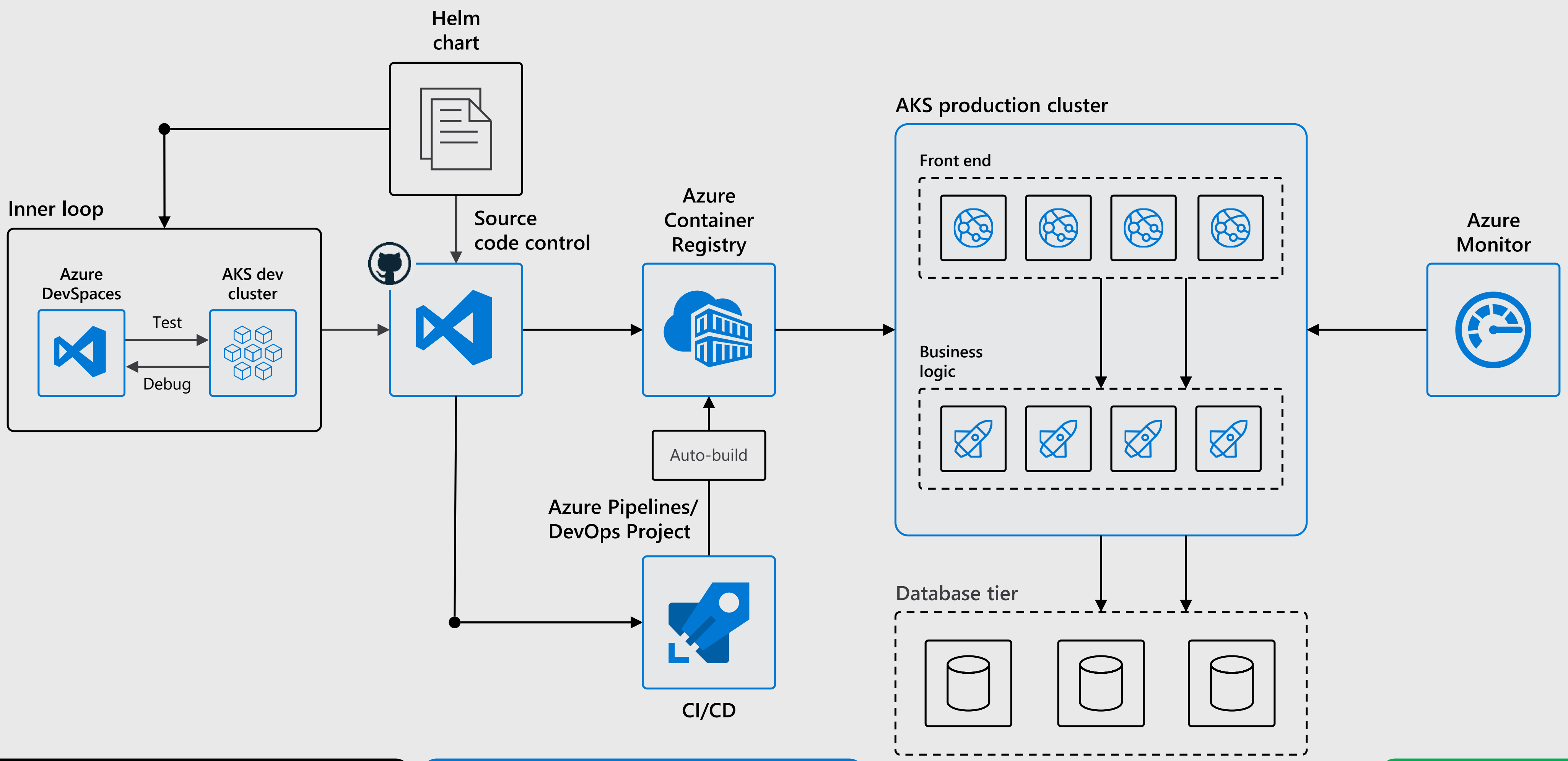
- NOW supports both Linux and Windows as Nodes

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
```

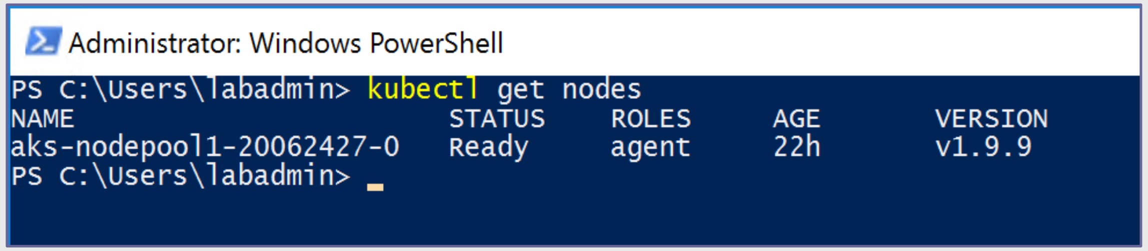
```
az aks get-credentials –resource-group AKSnativeRG –
name AKSCluster
```

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> _
```

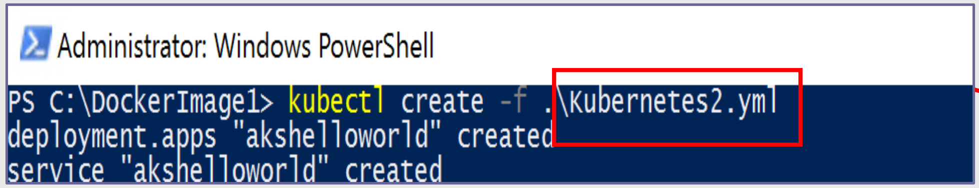
Demo

Deploying Azure Kubernetes Services using Azure CLI

Integrating AKS with ACR

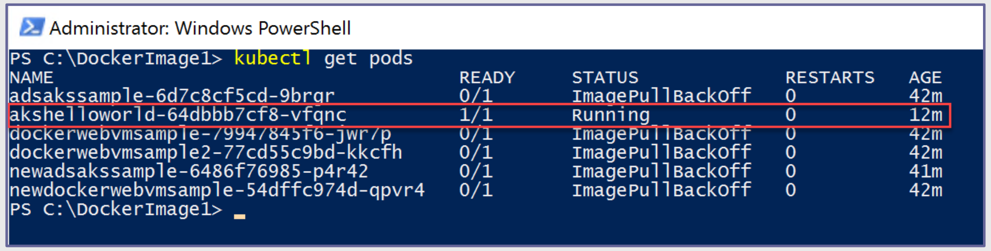
Running a Container Registry Image inside Azure Container Services

```
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

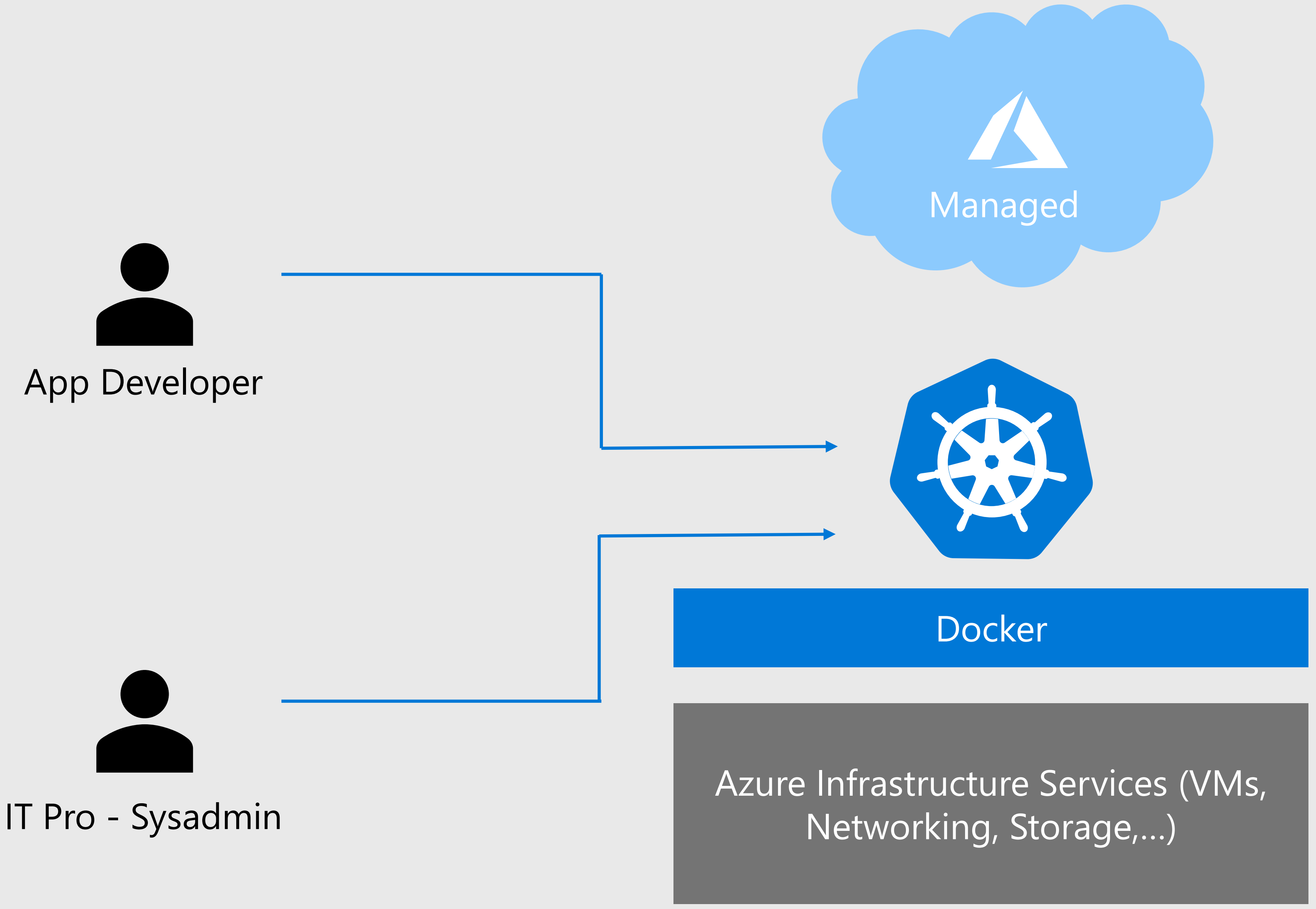
```
PS C:\DockerImage1> kubectl get pods
NAME                                  READY     STATUS             RESTARTS   AGE
adsakssample-6d7c8cf5cd-9brgr         0/1       ImagePullBackOff   0          42m
akshelloworld-64dbbb7cf8-vfqnc        1/1       Running            0          12m
dockerwebvmsample-79947845f6-jwr7p    0/1       ImagePullBackOff   0          42m
dockerwebvmsample2-77cd55c9bd-kkcfh   0/1       ImagePullBackOff   0          42m
newadsakssample-6486f76985-p4r42      0/1       ImagePullBackOff   0          41m
newdockerwebvmsample-54dffc974d-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

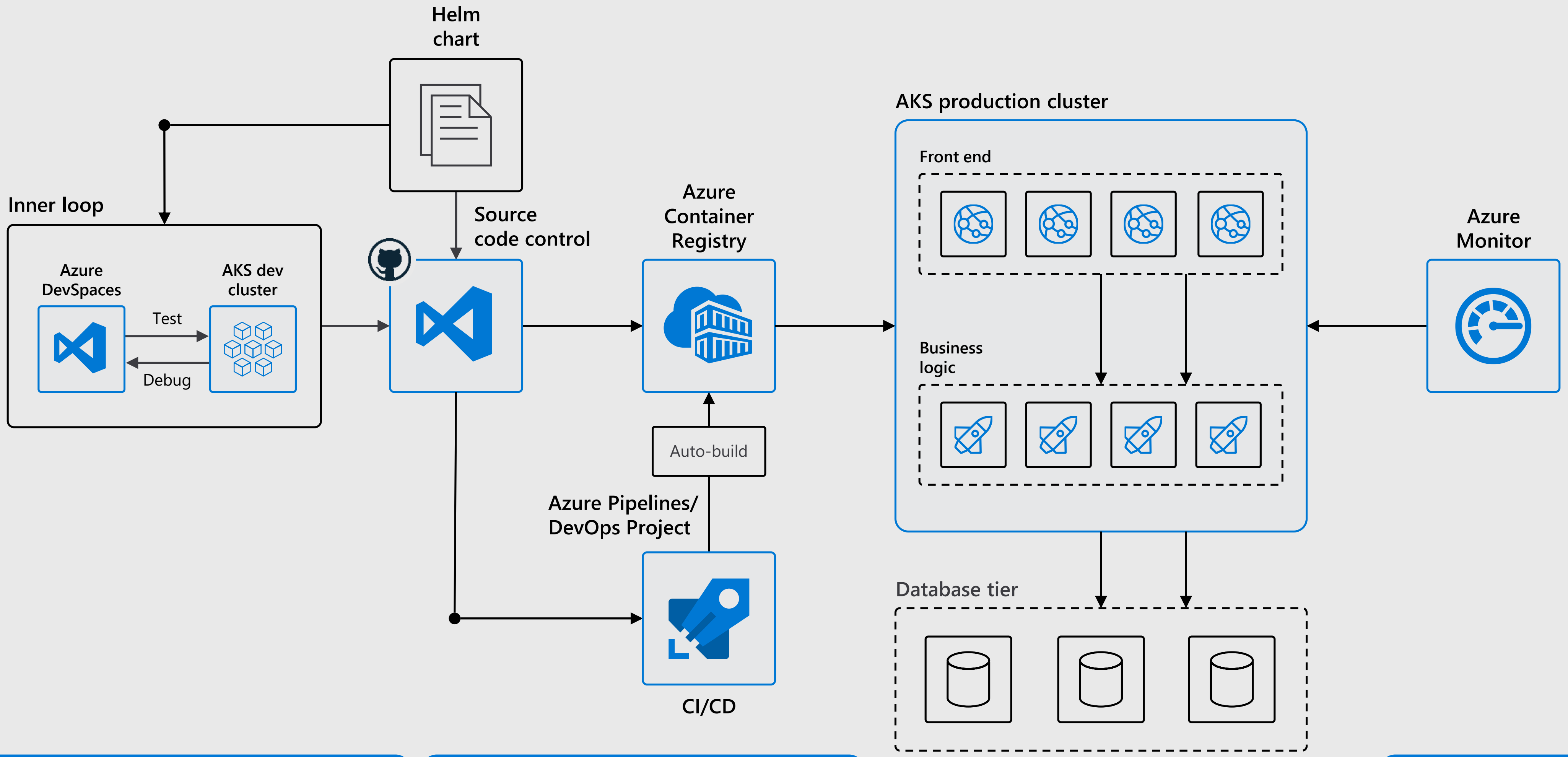
Running a Docker Hub image in Azure Kubernetes Services

Azure Kubernetes Service Architecture



- Managed Control Plane
- Automated upgrades and patching
- Easy cluster scaling
- Cost Savings

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

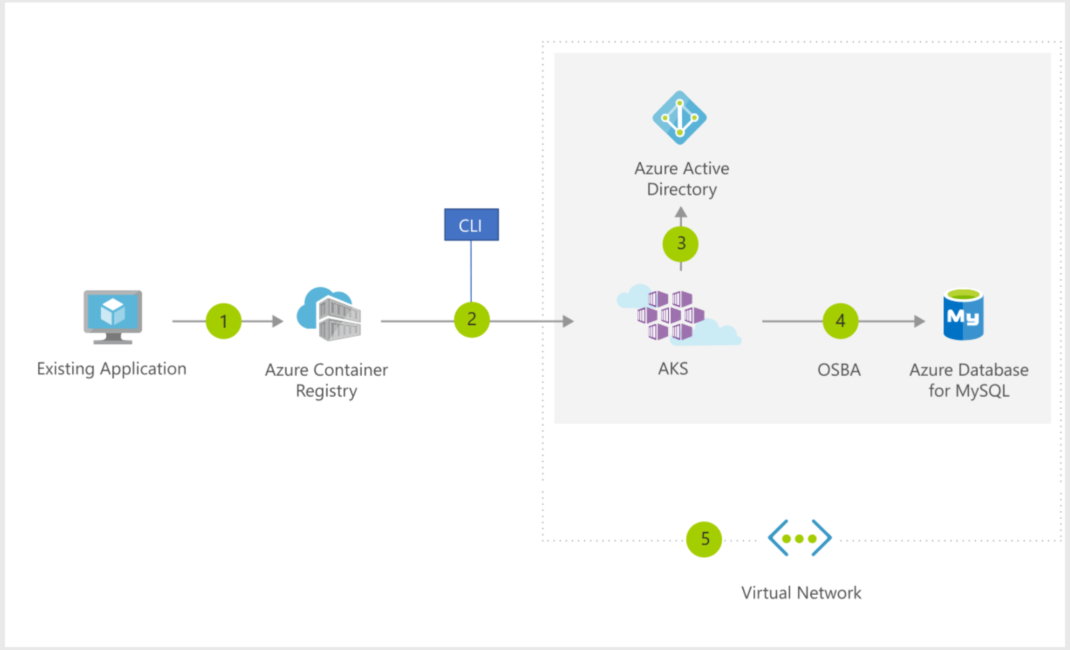
Azure Dev Spaces with AKS

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

Lab

Deploying Azure Container Services with Kubernetes

Deploying Azure Kubernetes Services

Lab 6 and 7 – Quick Instructions

1. (Assumption is you finished all prior labs to these)
2. Download the "Lab 6" and "Lab 7" Guide from GitHub (PDF)
3. Task 1: Deploy ACS (lab 6) or AKS (lab 7) from Azure CloudShell
4. Task 2: Manage ACS (lab 6) or AKS (lab 7) cluster using kubectl
5. Task 3: Integrating ACR with ACS (lab 6) or AKS (lab 7)
6. When having questions: msdevseriessupport@007FFFLearning.com

Section Take-Aways

1. Azure offers multiple Container Orchestrator solutions
2. Azure leverages on Kubernetes as the « go-to » Container solution, offering ACS and AKS
3. ACS will fade out => AKS (no Windows container support yet…)

Microsoft

Azure

Questions?

Peter De Tender

@pdtit @007FFFlearning

Microsoft

Azure

Next Module…

Azure Container Services Monitoring

Peter De Tender

@pdtit @007FFFlearning