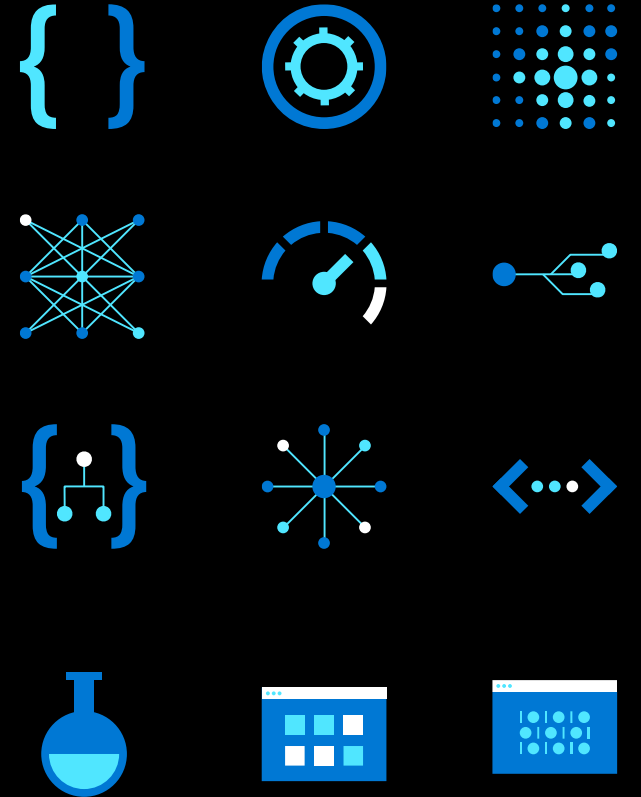


Azure Training Day

Run cloud-native apps with Azure Kubernetes Service



Optimize operations and monitoring

Part 3 of 4 in the Run cloud-native apps with Azure Kubernetes Service series

About us...

Ryan Berry

 Principal 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
- Module 3: Optimizing Azure Operations and Monitoring **YOU ARE HERE**
- 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**
 - *Lab 6: Deploying Azure Kubernetes Services and running containerized apps from Azure Container Registry*
- **Module 3: Optimizing Azure Operations and Monitoring **YOU ARE HERE****
 - *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.

Key Objectives

What you will learn in this section

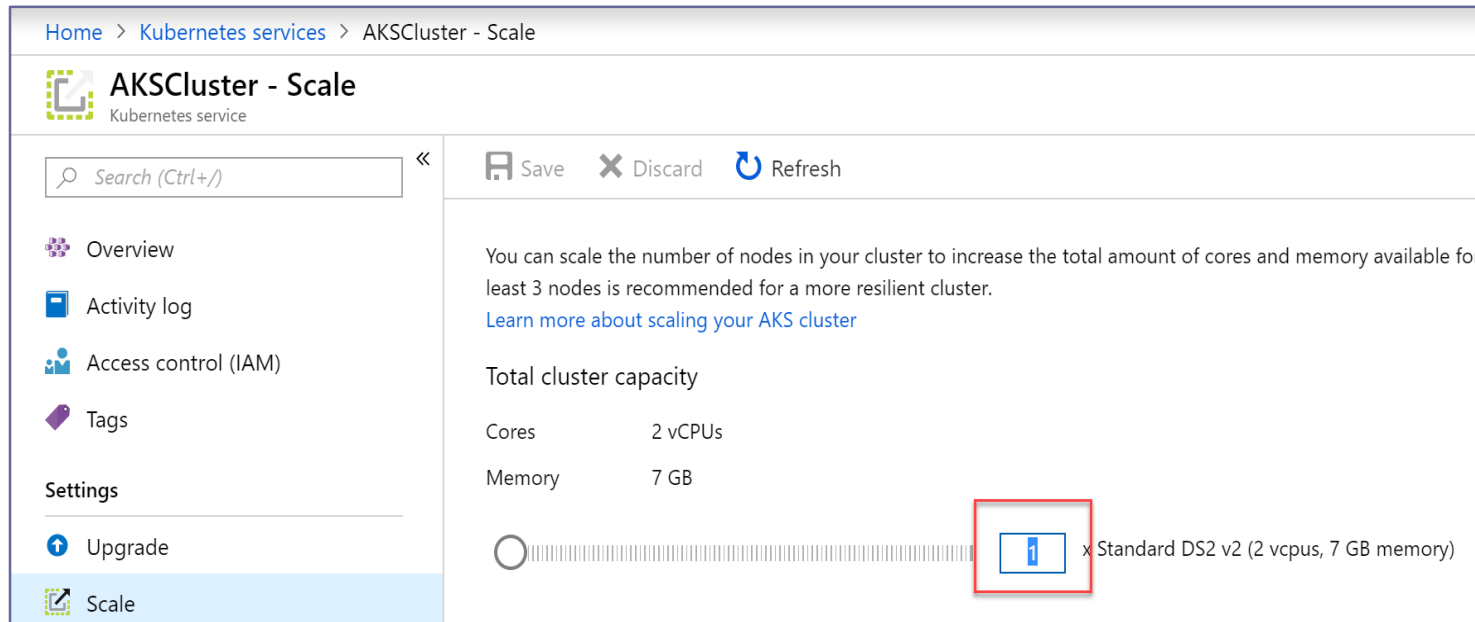
- Enabling Container Scalability in AKS
- Monitoring Azure Kubernetes Services
- Using Kubernetes dashboard with Azure Kubernetes Services

Enabling Container Scalability

AKS Scale

Option 1: Scaling the Nodes

Az **aks** scale --resource-group AKSRG --name AKSCluster --node-count 3



The screenshot displays the 'AKSCluster - Scale' page in the Azure portal. The breadcrumb navigation at the top reads 'Home > Kubernetes services > AKSCluster - Scale'. The page title is 'AKSCluster - Scale' with the subtitle 'Kubernetes service'. On the left sidebar, there is a search bar and a list of navigation options: Overview, Activity log, Access control (IAM), Tags, Settings, Upgrade, and Scale (which is currently selected). The main content area includes a 'Save', 'Discard', and 'Refresh' action bar. Below this, a message states: 'You can scale the number of nodes in your cluster to increase the total amount of cores and memory available for your cluster. At least 3 nodes is recommended for a more resilient cluster. [Learn more about scaling your AKS cluster](#)'. The 'Total cluster capacity' section shows 'Cores 2 vCPUs' and 'Memory 7 GB'. At the bottom, a progress bar is shown with a red box highlighting the number '1' in a blue box, indicating the current node count. To the right of the progress bar, the text 'x Standard DS2 v2 (2 vcpus, 7 GB memory)' is visible.

AKS Scale

Option 2: Scaling the PODS

```
kubectl scale --replicas=3 -f .\kubernetes3.yml
```

```
Administrator: Windows PowerShell
PS C:\Users\labadmin> cd\
PS C:\> cd .\DockerImage1\
PS C:\DockerImage1> kubectl scale --replicas=3 -f .\kubernetes3.yml
deployment.apps "drupalcntr" scaled
error: Scaling the resource failed with: could not fetch the scale for services drupalcntr
e requested resource
PS C:\DockerImage1> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
adsakssample-6d7c8cf5cd-9brgr	0/1	ImagePullBackoff	0	22h
akshellworld-64dbbb7cf8-vfqnc	1/1	Running	1	21h
dockerwebvmsample-79947845f6-jwr7p	0/1	ImagePullBackoff	0	22h
dockerwebvmsample2-77cd55c9bd-kkcfh	0/1	ImagePullBackoff	0	22h
drupalcntr-5fff4774bf-h8bx2	1/1	Running	0	20s
drupalcntr-5fff4774bf-hdm6g	1/1	Running	0	20s
drupalcntr-5fff4774bf-zm8lk	1/1	Running	0	50m
newadsakssample-6486f76985-p4r42	0/1	ImagePullBackoff	0	22h
newdockerwebvmsample-54dfc974d-qpv4	0/1	ImagePullBackoff	0	22h
ubuntucont-6f555d84d8-xs7vl	0/1	CrashLoopBackoff	15	54m

```
PS C:\DockerImage1> kubectl get services --watch
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
adsakssample	LoadBalancer	10.0.212.10	104.209.177.162	80:30156/TCP	22h
akshellworld	LoadBalancer	10.0.164.32	137.116.72.252	80:31558/TCP	21h
drupalcntr	LoadBalancer	10.0.74.211	104.46.117.95	80:30750/TCP	50m
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	22h
newadsakssample	LoadBalancer	10.0.56.37	104.209.180.231	80:32692/TCP	22h
ubuntucont	LoadBalancer	10.0.254.169	104.210.11.189	80:31412/TCP	55m

Scaling PODS means
you technically deploy
multiple instances of
your app container

AKS Container AutoScaler

Preview

```
kubectl create -f aks-cluster-autoscaler.yaml
```

```
kubectl autoscale deployment azure-vote-front --cpu-percent=50 --min=3 --max=10
```

AutoScaling checks pending PODS, based on values in the Yaml-file.

Define a min and max value

e.g. Run 3 nodes minimum, with a maximum of 10, when CPU +50%.

```
memory: 300Mi
command:
- ./cluster-autoscaler
- --v=3
- --logtostderr=true
- --cloud-provider=azure
- --skip-nodes-with-local-storage=false
- --nodes=1:10:nodepool1
env:
- name: ARM_SUBSCRIPTION_ID
  valueFrom:
```

Demo

Scaling Azure Kubernetes Services

Monitoring AKS in Azure

AKS Monitoring

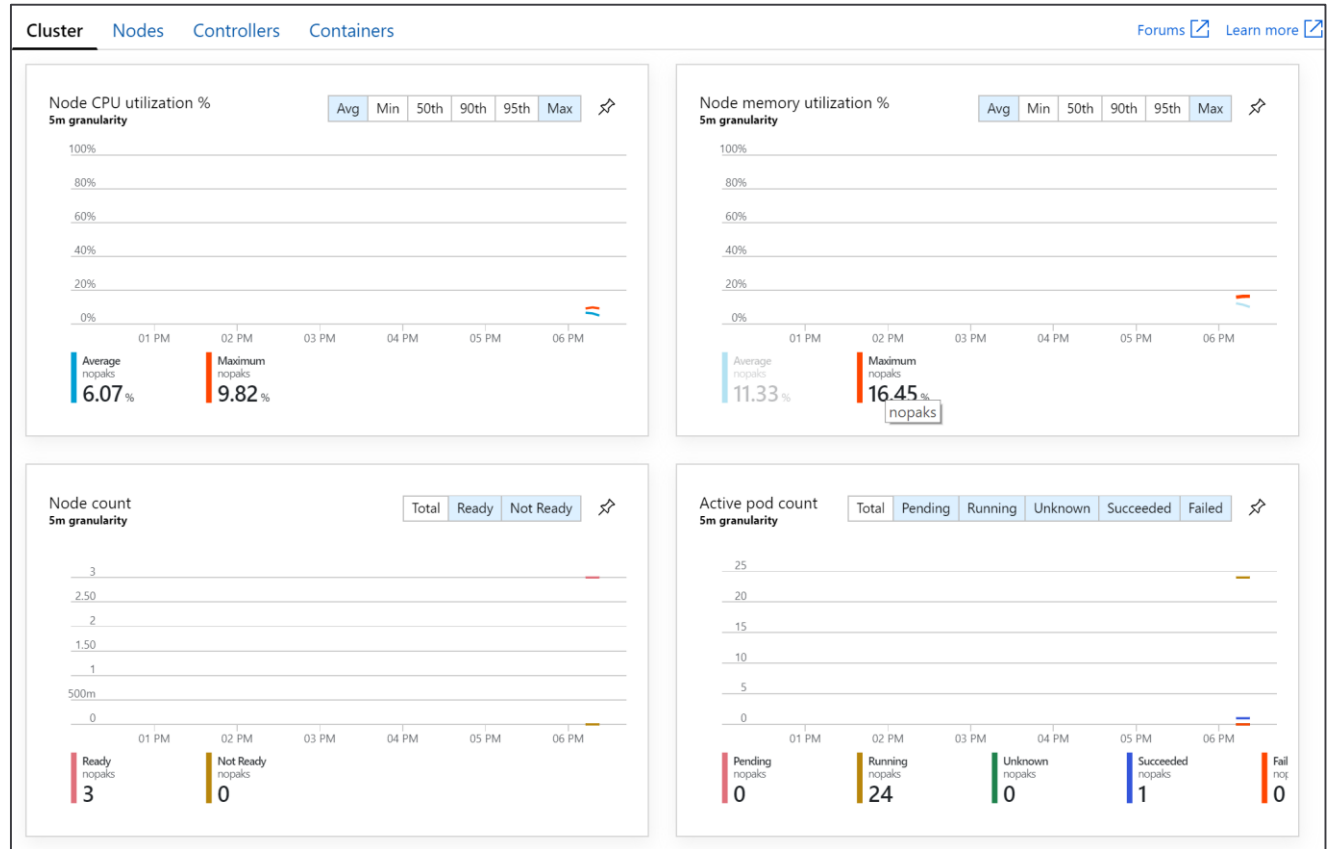
You can't manage what you don't see...

- Identify AKS containers that are running on the node and their average processor and memory utilization.
- Identify where the container resides in a controller or a pod.
- Review the resource utilization of workloads running on the host that are unrelated to the standard processes that support the pod.
- Understand the behavior of the cluster under average and heaviest loads.

AKS Monitoring

Built-in monitoring capabilities of the AKS Cluster, provided by Azure Monitor

- Cluster
- Nodes
- Controllers
- Containers



AKS Monitoring

Built-in monitoring capabilities of the AKS Cluster, provided by Azure Monitor

- Cluster
- Nodes
- Controllers
- Containers

Cluster Nodes Controllers Containers								
Search by name...			Metric: CPU Usage (millicores)		Min Avg 50th 90th 95th Max			
NAME	STATUS	95TH %↓	95TH	CONTAIN...	RESTA...	UPTIME	NODE	TREND 95TH
▶ kubernetes-dashboard-...	1	0.5%	0.5 mc	1	0	13 hours	-	
▶ kube-proxy (DaemonSet)	7	0.4%	7 mc	7	0	10 mins	-	
▶ kube-svc-redirect (Dae...	3	0.2%	3 mc	6	0	18 mins	-	
▶ kube-dns-v20-7d874cb...	2	0.1%	2 mc	6	0	13 hours	-	
▶ akshelloworld-64dbbb7...	1	0%	0.2 mc	1	0	13 hours	-	
▲ drupalcntr-5fff4774bf (R...	3	0%	0.1 mc	3	0	20 mins	-	
▲ drupalcntr-5fff4774...	Ok	0%	0.1 mc	1	0	1 hour	aks-nodepo...	
adsacr	Ok	0%	0.1 mc	1	0	1 hour	aks-nodepo...	
▲ drupalcntr-5fff4774...	Ok	0%	0.1 mc	1	0	20 mins	aks-nodepo...	
adsacr	Ok	0%	0.1 mc	1	0	20 mins	aks-nodepo...	
▲ drupalcntr-5fff4774...	Ok	0%	0.1 mc	1	0	20 mins	aks-nodepo...	
adsacr	Ok	0%	0.1 mc	1	0	20 mins	aks-nodepo...	

nopacr1
Container

View live data(preview)

View in analytics

Container Name
nopacr1

Container ID
4d310536ed49ec0112e482095809e826abf
10b3bf4595dcfaef08de0d13e8991

Container Status
running

Container Status Reason
-

Image
simplpdtv1

Image Tag
latest

Container Creation Time Stamp
8/23/2019, 6:12:14 PM

Start Time
8/23/2019, 6:12:14 PM

Finish Time
-

CPU Limit
1931 mc

CPU Request
0 mc

Memory Limit
4.45 GB

AKS Monitoring – Log Analytics

Run Powerful Queries against the AKS Cluster, provided by Log Analytics

- Cluster
- Nodes
- Controllers
- Containers

The screenshot displays the Azure Log Analytics 'Logs' page for a specific workspace. The breadcrumb navigation shows the path: Home > Kubernetes services > nopaks - Insights > Logs. The workspace ID is 'defaultworkspace-0a407898-c077-442d-8e17-71420aa82426-weu'. The interface includes a 'New Query 1*' button and a toolbar with options like Help, Settings, Sample queries, and Query explorer. The left sidebar shows the 'Schema' and 'Filter' tabs, with a search bar and a list of active workspaces. The main area contains a Kusto query to filter logs by container name and cluster ID. The query is as follows:

```
let startDateTime = datetime('2019-08-23T10:15:00.000Z');
let endDateTime = datetime('2019-08-23T16:27:10.310Z');
let ContainerIdList = KubePodInventory
| where TimeGenerated >= startDateTime and TimeGenerated < endDateTime
| where ContainerName =~ 'c7578dcb-c5c0-11e9-9f85-d29e4a9b2f68/tunnel-front'
| where ClusterId =~ '/subscriptions/0a407898-c077-442d-8e17-71420aa82426/resourceGroups/nopaksrc/providers/Microsoft.ContainerService/managedCluster'
| distinct ContainerID;
ContainerLog
| where TimeGenerated >= startDateTime and TimeGenerated < endDateTime
| where ContainerID in (ContainerIdList)
| project LogEntrySource, LogEntry, TimeGenerated, Computer, Image, Name, ContainerID
```

At the bottom, the status indicates 'Completed' with a duration of 00:00:02.435 and 0 records returned. View options for 'TABLE' and 'CHART' are available.

Demo

Monitoring AKS in Azure

Using the Kubernetes Dashboard

Kubernetes Dashboard

You can't manage what you don't see...

- Kubernetes includes a **web dashboard** that can be used for **basic management operations**. This dashboard lets you view basic health status and metrics for your applications, create and deploy services, and edit existing applications.

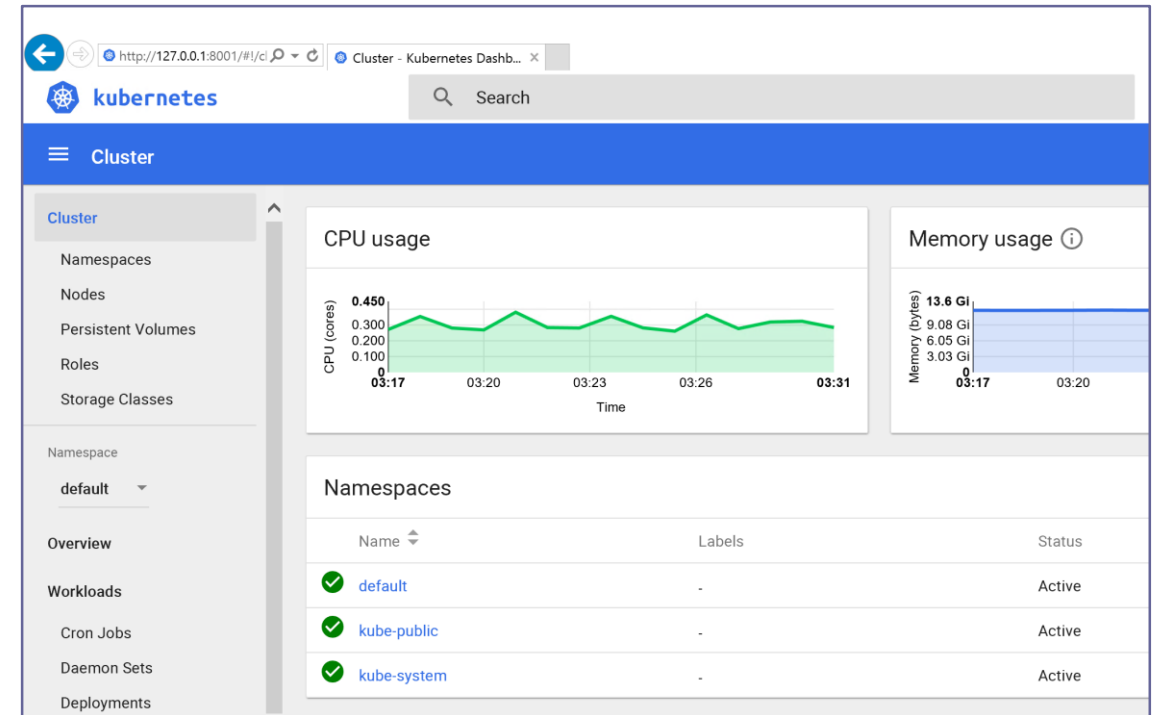
```
az aks browse --resource-group AKSResourceGroup --name AKSCluster
```


Using the Default Kubernetes Dashboard

You don't have to use the Azure Monitor-based version...

1. Install latest version of the kubectl CLI tool
2. Connect to the AKS Cluster with the correct credentials
3. `az aks browse...`

Could be beneficial in a multi-environment Kubernetes setup, as the dashboard is native to Kubernetes, no matter where the cluster runs



Using the Default Kubernetes Dashboard

You don't have to use the Azure Monitor-based version...

Shows detailed information on:

- Jobs
- Pods
- Nodes
- ...

<div>Namespace</div> <div>default</div> <div>Overview</div> <div>Workloads</div> <div>Cron Jobs</div> <div>Daemon Sets</div> <div>Deployments</div> <div>Jobs</div> <div>Pods</div> <div>Replica Sets</div>	✓	drupalcntr-5fff4774bf-h8b	aks-nodepool1-20062427-0	Running	0	43 minutes	<div></div>	0
	✓	drupalcntr-5fff4774bf-hdm	aks-nodepool1-20062427-0	Running	0	43 minutes	<div></div>	0
	✓	drupalcntr-5fff4774bf-zm8	aks-nodepool1-20062427-0	Running	0	an hour	<div></div>	0
	!	ubuntucont-6f555d84d8-x	aks-nodepool1-20062427-0	Waiting: CrashLoop	24	an hour	-	
		Back-off restarting failed container						
	✓	akshelloworld-64dbbb7cf8	aks-nodepool1-20062427-0	Running	1	22 hours	<div></div>	0
	!	newadsakssample-6486f7	aks-nodepool1-20062427-0	Waiting: ImagePullE	0	23 hours	-	
		Error: ImagePullBackOff						

If AKS cluster is using RBAC, your dashboard will fail because of missing rolebinding:



```
kubectl create clusterrolebinding kubernetes-  
dashboard --clusterrole=cluster-admin --  
serviceaccount=kube-system:kubernetes-dashboard
```

Demo

Using the Kubernetes Dashboard

Section Take-Aways

1. Azure Kubernetes Services has built-in scaling features
2. Azure Kubernetes Services provides Azure-integrated Monitoring, relying on Azure Monitor and Log Analytics
3. AKS provides support to open the traditional« Kubernetes Dashboard »

Overall Workshop Take-Aways

1. Azure has all services available to assist in application migration and digital transformation
2. Azure integrates with Docker Containers, as an enabler for cloud migration and integration of your business-critical workloads
3. Azure provides several « Container Services », from low-level to enterprise-grade solutions, including end-to-end monitoring, relying on Azure common services



Azure

Questions?

Peter De Tender

Thank you for having attended this workshop



Peter De Tender



Thank You