GitHub & DevOps – using AKS

PROBLEM STATEMENT :

In today's rapidly changing business landscape, traditional application deployment falls short in meeting the growing demands for scalability and cross-platform compatibility. To address these challenges, a DevOps solution combining GitHub Actions and Azure Kubernetes Services (AKS) is crucial. This solution automates container management, ensuring efficient, secure, and consistent application deployment across diverse environments.

SOLUTION ARCHITECTURE :



Prerequisites:

- Access to an Azure subscription
- Access to a GitHub account
- Basic knowledge of executing commands by using the Azure CLI
- Basic knowledge of Kubernetes and its concepts
- Basic knowledge of AKS and its concepts
- Basic knowledge of Git and GitHub

Step 1: Creating a Kubernetes cluster in azure portal.

Before creating a Kubernetes cluster you need to create a resource group in

which your cluster will reside

Home 2 Resource groups 2						
Resource groups Default Directory (Cloudbeaver net)	4	Resource group				×
🕂 Create 🔘 Manage view 🗸 \cdots		P Search (Ctrl+/)	+ Add \vee 💷 Edit columns 🔋 Delete resource group	🖒 Refresh 🛓 Export	to CSV 🧏 Open que	y
Filter for any field	-1	Overview	↑ Essentials			250N View
Name 1		Activity log	Subscription (change) Microsoft Partner Network	Deployments 1 Deploying		
(a) AKS_RG (a) DefaultResourceGroup-EUS (b) MC_AKS_RG_My_Kube_Cluster_eastus	•••	Access control (AM) Tags Events Settings	Subscription ID 1394a106-cfe6-4dbb-9b7a-574642464238 Tegs (change) Click here to add tags Filter for any field	Location East US on == all X *y Add filter		
		 Deployments Security 	Showing 1 to 1 of 1 records. Show hidden types ③	No grouping	V List view	Y
		Properties	C & the Kube Cluster	Kubernetes service	East US	

When setting up your cluster, the critical step is selecting the right node size and determining the number of nodes required. The node count represents the VMs within your cluster, with a maximum limit of 1000 nodes. Remember that you can adjust the node count post-deployment, but the instance type remains fixed. Keep in mind that if you opt for Linux as the OS, hosting Windows applications won't be possible, and vice versa.

🔲 Microsoft Azure 😕	earch resources, services, and docs (G+/)		E	₽
Home > Create a resource >				
Create Kubernetes c	luster			
	GREET NOW			
Cluster details				
Kubernetes cluster name * 💮	My_Kube_Cluster	4		
Region * ①	(US) East US	~		
Availability zones 🛈	Zones 1,2,3	~		
Kubernetes version * ()	1.19.9 (default)	~		
Primary node pool				
The number and size of nodes in the recommended for resiliency. For dev additional node pools or to see addit be able to add additional node pools	primary node pool in your cluster. For production elopment or test workloads, only one node is requir ional configuration options for this node pool, go after creating your cluster. Learn more about noc	workloads, at least 3 nodes are red. If you would like to add to the 'Node pools' tab above. You will de pools in Azure Kubernetes Service		
Node size * 🛞	Standard DS2 v2 Change size			
Node count * ①	0	3		

MC_AKS_RG_M	ly_Kube_Cluster_eastus 🖈 🐃		×			
P Search (Ctrl+/)	🔹 🕂 Add 🗸 III Edit columns 🔋 Delete resource group 🖒 Refresh	🛓 Export to CSV 😤 Open query 😋	Feedback 📮 Open in mobile			
W Overview	° ✓ Essentials	* V Essentials JSON Vie				
Activity log	Filter for any field	* _▼ Add filter				
R Access control ((AM)	Showing 1 to 7 of 7 records.	Nog	rouping 🗸 Uist view 🗸			
 Formers 	Name †.	Type τ_{\perp}	Location \uparrow_{Φ}			
fattings	aks-agentpool-27754862-nsg	Network security group	East US			
seconds	🔤 🐁 aks-agentpool-27754882-rootetable	Route table	East US			
- Deployments	🗾 🗞 aks-agentpool-27734862-vmss	Virtual machine scale set	East US			
Security		Virtual network	East US			
Policies	E 16d7e85c-3d97-4aa8-96e3-fc1d4cb137e4	Public IP address	East US			
Properties	Viubemetes	Load balancer	East US			
🔒 tocks	My_Kube_Cluster-agentpool	Managed Identity	East US			

You can also use CLI :

Add windows server 2019 node pool:

Ex:

az aks nodepool add $\ *$ /used to add a node pool named npwin

```
--resource-group myResourceGroup \
```

```
--cluster-name myAKSCluster \
```

```
--os-type Windows \
```

```
--name npwin \
```

--node-count 1

Add a windows server node pool 2022:

For newer versions we need to specify os sku type.

Ex:

```
az aks nodepool add \
```

- --resource-group myResourceGroup \
- --cluster-name myAKSCluster \
- --os-type Windows \
- --os-sku Windows2022 $\$
- --name npwin \

--node-count 1

connect to the cluster:

• To connect to the cluster we use the kubectl it can installed locally using the

command az aks install-cli.az aks get-credentials --resource-group

myResourceGroup --name myAKSCluster is the command used to configure

kubectl.

To check the status of each node execute the following command:

kubectl get nodes -o wide

Step 2 :

In Azure CLI clone the github repository



cd mslearn-aks-deployment-pipeline-github-actions Bash bash init.sh	Bash					
Bash bash init.sh	cd mslearn-aks-deployment-pipeline-github-actions					
Bash bash init.sh						
bash init.sh	Dk					
bash init.sh	Bash					
	bash init.sh					

Step 3:

Build the Application WorkFlow



Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow template to get started. Skip this and set up a workflow yourself \Rightarrow

Step 4:

Create a YAML file for pushing and deploying

name: Build and push the latest build to staging

on:

push:

branches: [main]

jobs:

build_push_image:

runs-on: ubuntu-20.04

steps:

- uses: actions/checkout@v2

name: Set up Buildx
 uses: docker/setup-buildx-action@v1

- name: Docker Login uses: docker/login-action@v1 with: registry: \${{ secrets.ACR_NAME }} username: \${{ secrets.ACR_LOGIN }} password: \${{ secrets.ACR_PASSWORD }}

name: Build and push staging images
 uses: docker/build-push-action@v2

with:

context: .

push: true

tags: \${{secrets.ACR_NAME}}/contoso-website:latest

Commit the Changes:

Create build	steps for staging	
Add an optiona	l extended descrip	tion…
oose which email a	ddress to associate with t	¢
) -O- Commit dire	ectly to the main brar	nch.
) ያጎ Create a ne pull request. Le	w branch for this com arn more about pull re	mit and star quests.

Step 5:

Connect the AKS CLI

\$ kubect1 get nodes				
NWE	STATUS	ROLES	AGE	VERSION
aks-agentpool-27754862-vmss800000	Ready	agent	3m31s	v1.19.9
aks-agentpool-27754862-vmss808001	Ready	agent	3 n 47s	v1.19.9

Step 6 :

apiVersion: apps/v1

kind: Deployment

metadata:

name: sam

labels:

app: sam

spec:

replicas: 1

template:

metadata:

name: sam

labels:

app: sam

spec:

nodeSelector:

"kubernetes.io/os": windows

containers:

- name: sam

image: E:/dotnet/framework/sam:asp netapp

resources:

limits:

cpu: 1

memory: 800M

ports:

- containerPort: 80

selector:

matchLabels:

app: sam

apiVersion: v1

kind: Service

metadata:

name: sam

spec:

type: LoadBalancer

ports:

- protocol: TCP

port: 80

selector:

app: sample

Deploy the application using kubectl apply:

kubectl apply -f sam.yaml

Your application is deployed.

Docker Configuration:

search for **Docker Login**. Select the first result published by **Docker**.

Build and push docker images

Marketplace / Search results



Build and push Docker images By docker 🔗 Build and push Docker images with Buildx ☆ 733

Create a personal access token (PAT)

- 1. Go to the fork of the sample repository in the GitHub website. On the top right hand corner, select your profile photo, then select **Settings**.
- 2. Select **Developer settings** at the bottom of the left menu.
- 3. Select Personal access tokens.
- 4. Select Generate new token.
- 5. Provide a name for your PAT, such as myPersonalAccessToken
- 6. Select the checkbox next to **public_repo**.

Search or jump to	/ Pull requests issues	Marketplace Explore				
Settings / Developer settings						
器 GitHub Apps	New personal acc	ess token				
A OAuth Apps Personal access tokens	Personal access tokens function HTTPS, or can be used to auth	Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.				
	Note					
	myPersonalAccessToken					
	What's this token for?					
	Expiration *	Expiration *				
	7 days	will expire on Wed, Apr 13 2022				
	Select scones					
	Scopes define the access for p	ersonal tokens. Read more about OAuth scopes.				
	🗆 repo	Full control of private repositories				
	repo:status	Access commit status				
	repo_deployment	Access deployment status				
	public_repo	Access public repositories				
	repositivite	Access repository invitations				
	security_events	Read and write security events				

Install Helm

In this exercise, you use Helm version v3.3.1. Azure has a built action that downloads and installs Helm.

1. Below the runs-on key, add a new steps key. Then, search for **Helm tool installer**. Select the first result published by **Azure**.

deploy:

runs-on: ubuntu-20.04

needs: build_push_image

steps:

- uses: actions/checkout@v2

- name: Helm tool installer

uses: Azure/setup-helm@v1

with:

Version of helm

version: # default is latest

Marketplace / Search results / Helm tool installer

Helm tool installer By Azure 📀 🔊 v3.3 😭 77

Install a specific version of helm binary. Acceptable values are latest or any semantic version string like 1.15.0

View full Marketplace listing

Installation

Copy and paste the following snippet into your .yml file.

Version: v3.3 -

- 1. To commit the changes, select the **Start commit** button. Enter a description for the commit, and then select **Commit new file**.
- 2. In Cloud Shell, run git pull to fetch the latest changes. Then, run the following command to tag and push the changes:

BashCopy

git tag -a v2.0.1 -m 'Creating first production deployment' && git push --tags

- 3. When prompted, provide your GitHub username, and the PAT created previously as the password.
- 4. Open the **Actions** tab and see the running process.

CHALLENGES FACED:

- 1. Initial Configuration Complexity: Setting up AKS to meet project requirements initially posed a challenge. Detailed documentation from Microsoft and reference materials from various GitHub repositories were crucial resources.
- 2. **Multi-Cloud Deployment Complexity:** Deploying applications across multiple cloud infrastructures increases the complexity of managing Kubernetes clusters. Coordinating networking and storage across different environments can be challenging.

- 3. **Storage Management:** Handling storage can be a significant challenge, especially without relying on a cloud service. Kubernetes requires efficient storage solutions, and configuring this can be complex.
- 4. **Security Concerns:** Kubernetes is an open-source tool, which means security may not be as robust as desired. Ensuring the security of containerized applications and the cluster itself can be a challenge.
- 5. **Tooling and Interface Transition:** Transitioning from one environment to another, such as moving from CLI to PowerShell, can be challenging. Different environments offer varying features and require adaptation, adding complexity to DevOps processes

BUSINESS BENEFITS

Kubernetes offers robust load balancing capabilities for efficient traffic distribution.

It facilitates declarative configuration and automation, reducing the risk of human errors in managing complex deployments.

Implementing CI/CD with GitHub Actions and AKS brought several significant business benefits. It streamlined and automated the development pipeline, reducing manual image builds and deployments, which saved valuable time. This increased efficiency translated into cost savings, as the company no longer needed to allocate resources for time-consuming manual tasks. Furthermore, the solution improved the overall agility of the development process, enabling quicker feature releases and enhancing the company's competitiveness in the market.