

# Deploying a Simple Web Application on Azure Kubernetes Service (AKS)

---

## 1. Task Description

### Deploying a Simple Web Application on Azure Kubernetes Service (AKS)

The task involves creating all required Azure resources from scratch, containerizing a simple web application using Docker, storing the image in Azure Container Registry (ACR), deploying the application on Azure Kubernetes Service (AKS) using Kubernetes manifests, and exposing the application to the internet using a Kubernetes LoadBalancer service. The objective is to understand end-to-end container orchestration, Kubernetes core components, and Azure-managed Kubernetes services.

---

## 2. What You Will Build

You will deploy:

- A simple web application (NGINX with sample HTML)
  - Containerized using Docker
  - Image stored in Azure Container Registry (ACR)
  - Application deployed on Azure Kubernetes Service (AKS)
  - Application exposed publicly using Azure Load Balancer
- 

## 3. High-Level Architecture

User Browser



Azure Load Balancer (Kubernetes Service)



AKS Pods (NGINX Containers)



Azure Container Registry (Docker Image)

---

#### 4. Prerequisites and Why They Are Needed

Item	Purpose
Azure Account	To create and manage cloud resources
Azure Portal	GUI-based resource creation
Azure CLI	Command-line interaction with Azure
Docker	To containerize the web application
kubectl	To manage Kubernetes resources
Cloud Shell	Avoids local setup issues

---

#### 5. Step-by-Step Implementation (END TO END)

---

##### STEP 1: Login to Azure

##### Command (CLI / Cloud Shell)

**az login**

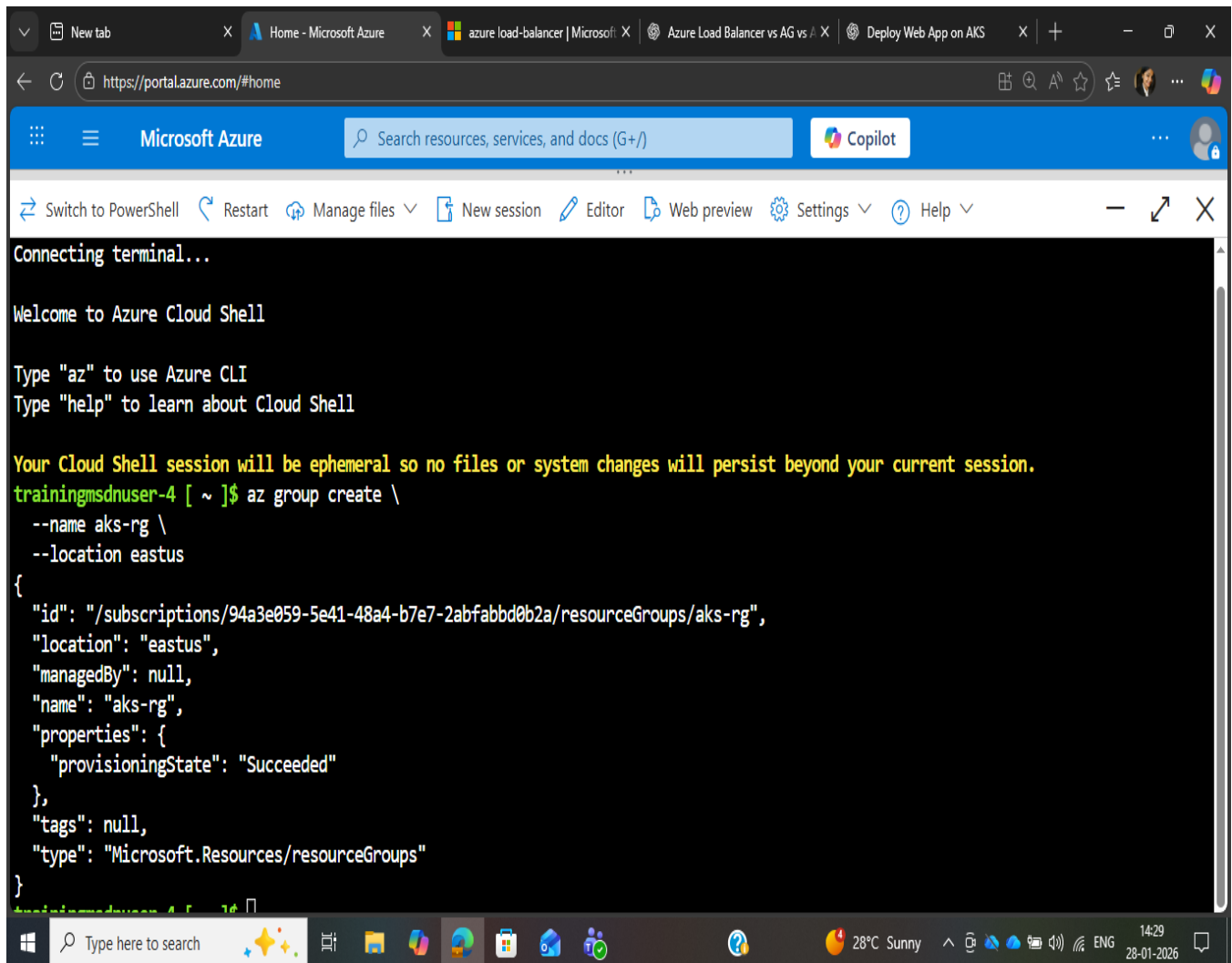
##### Purpose

- Authenticates the user with Azure
  - Grants permission to create resources in the subscription
- 

##### STEP 2: Create Resource Group

##### Command

**az group create --name aks-rg --location eastus**



## Purpose

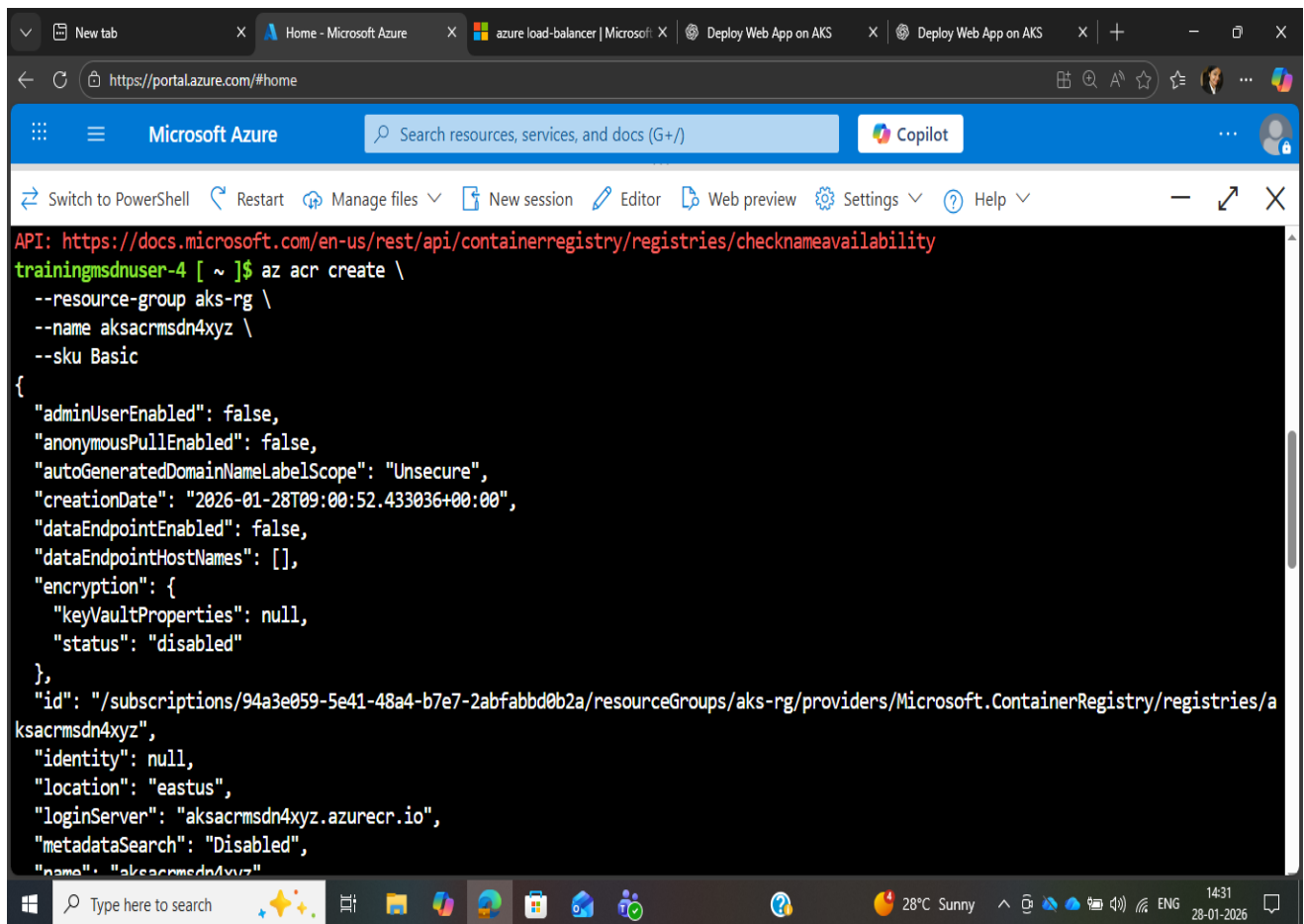
- **Resource Group is a logical container**
- **Helps in organizing, managing, and deleting resources together**
- **Deleting the resource group deletes all resources inside it**

---

## STEP 3: Create Azure Container Registry (ACR)

### Command

**az acr create --resource-group aks-rg --name aksacrmsdn4xyz --sku Basic**



The screenshot shows a terminal window within the Microsoft Azure portal. The terminal is running the command `az acr create` with the following flags: `--resource-group aks-rg`, `--name aksacrmsdn4xyz`, and `--sku Basic`. The output is a JSON object representing the created container registry. The API endpoint for checking name availability is also shown at the top of the terminal.

```
API: https://docs.microsoft.com/en-us/rest/api/containerregistry/registries/checknameavailability
trainingmsdnuser-4 [ ~ ]$ az acr create \
  --resource-group aks-rg \
  --name aksacrmsdn4xyz \
  --sku Basic
{
  "adminUserEnabled": false,
  "anonymousPullEnabled": false,
  "autoGeneratedDomainNameLabelScope": "Unsecure",
  "creationDate": "2026-01-28T09:00:52.433036+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/94a3e059-5e41-48a4-b7e7-2abfabbd0b2a/resourceGroups/aks-rg/providers/Microsoft.ContainerRegistry/registries/aksacrmsdn4xyz",
  "identity": null,
  "location": "eastus",
  "loginServer": "aksacrmsdn4xyz.azurecr.io",
  "metadataSearch": "Disabled",
  "name": "aksacrmsdn4xyz"
}
```

## Purpose

- Stores Docker images privately
- More secure than public registries
- AKS pulls application images from ACR

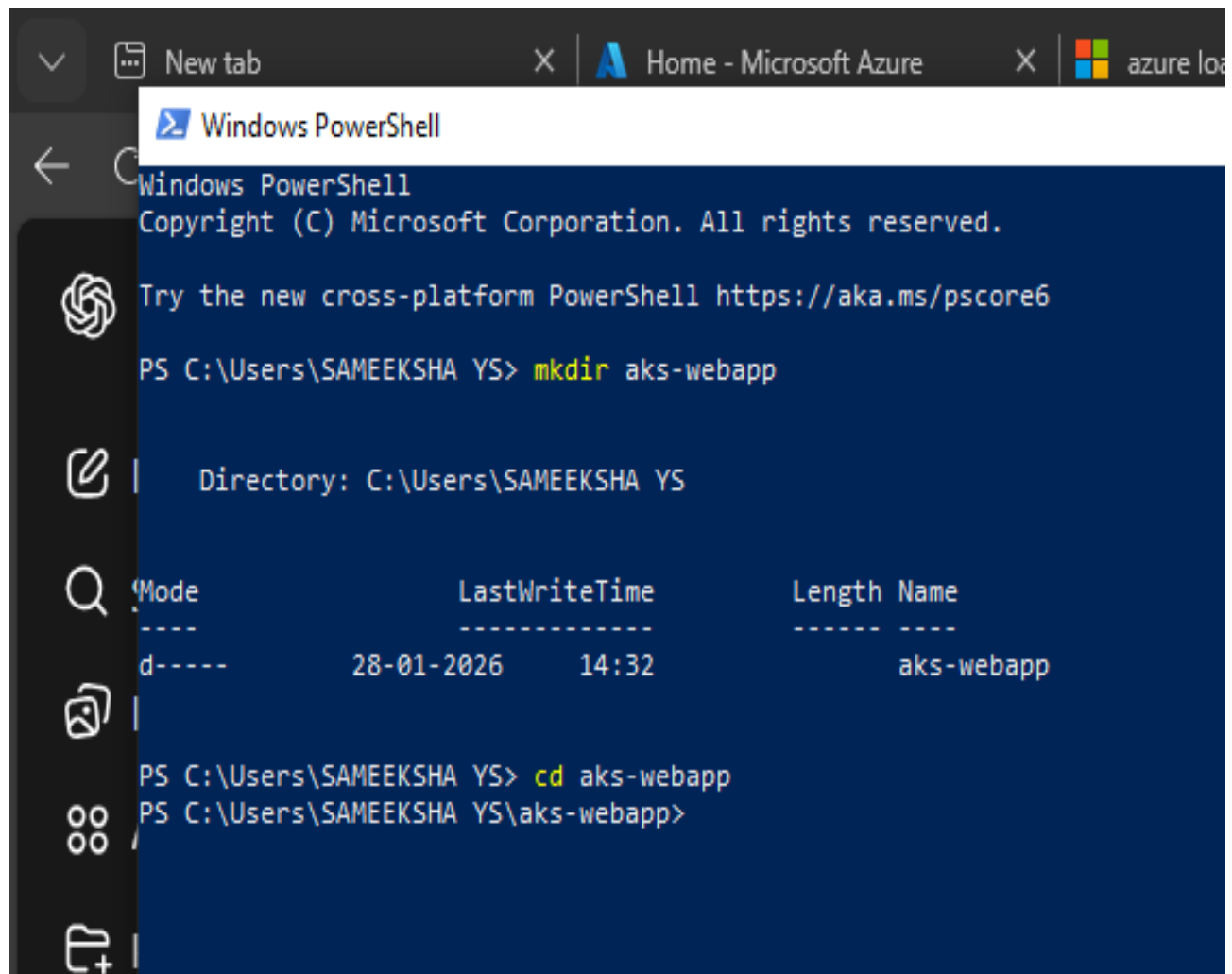
---

## STEP 4: Build the Web Application Locally

### Create Project Folder

```
mkdir aks-webapp
```

```
cd aks-webapp
```



The screenshot shows a Windows PowerShell terminal window. The title bar indicates it's a 'New tab' in a browser, with the address bar showing 'Home - Microsoft Azure'. The PowerShell prompt shows the user is in the directory 'C:\Users\SAMEEKSHA\YS'. The user has executed the command 'mkdir aks-webapp', which has created a new directory. The terminal then shows the directory listing for 'aks-webapp', which is currently empty. The user then navigates into the 'aks-webapp' directory using the 'cd' command.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\SAMEEKSHA\YS> mkdir aks-webapp

Directory: C:\Users\SAMEEKSHA\YS

Mode                LastWriteTime         Length Name
----                -
d-----          28-01-2026         14:32         aks-webapp

PS C:\Users\SAMEEKSHA\YS> cd aks-webapp
PS C:\Users\SAMEEKSHA\YS\aks-webapp>
```

Create index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>AKS Web App</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Hello from Azure Kubernetes Service!</h1>
```

```
</body>
```

```
</html>
```

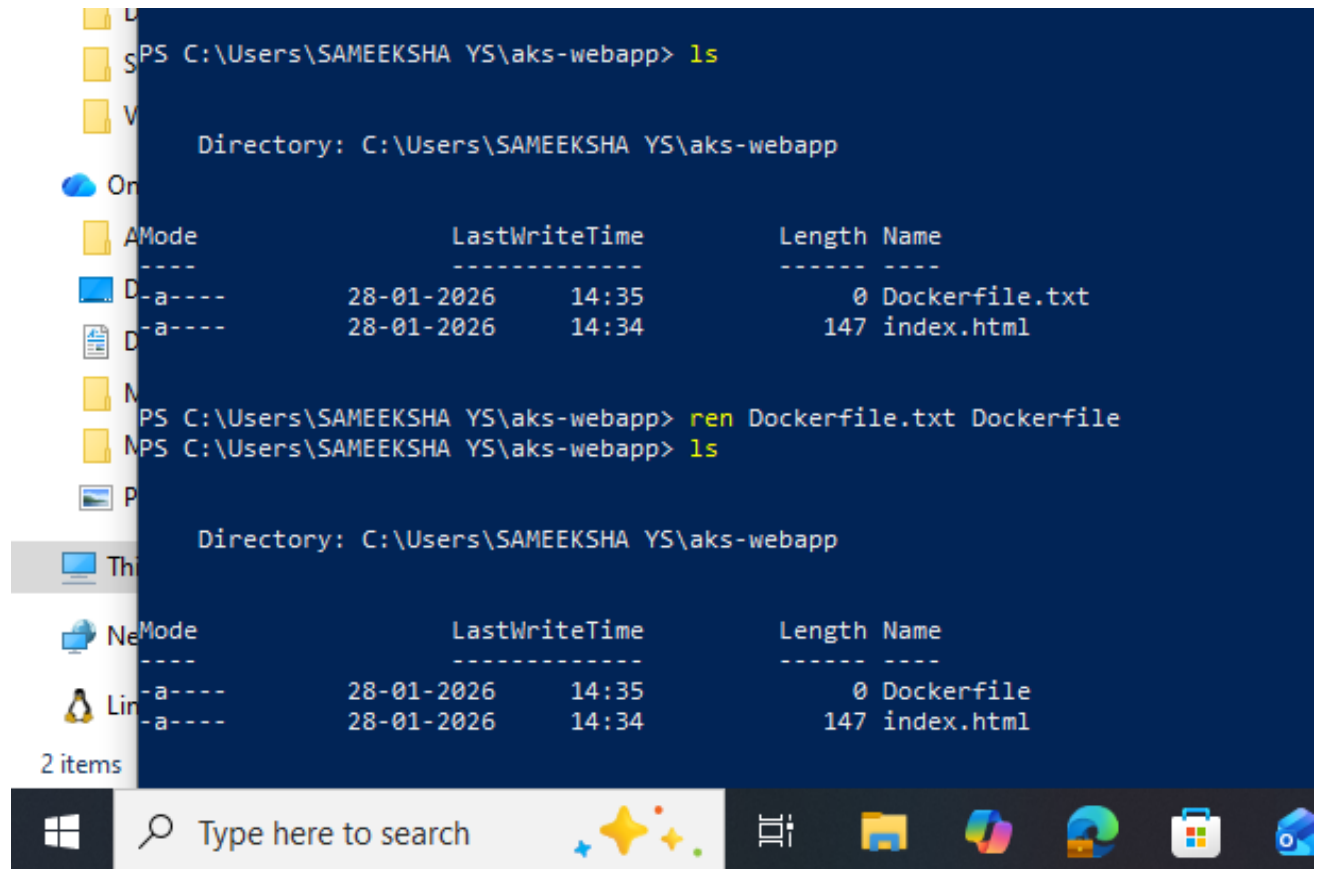
## Purpose

- Simple application to focus on AKS concepts
  - Avoids unnecessary coding complexity
- 

## STEP 5: Create Dockerfile

FROM nginx:latest

COPY index.html /usr/share/nginx/html/index.html



```
PS C:\Users\SAMEEKSHA YS\aks-webapp> ls

Directory: C:\Users\SAMEEKSHA YS\aks-webapp

Mode                LastWriteTime         Length Name
----                -
-a-----         28-01-2026         14:35             0 Dockerfile.txt
-a-----         28-01-2026         14:34          147 index.html

PS C:\Users\SAMEEKSHA YS\aks-webapp> ren Dockerfile.txt Dockerfile
PS C:\Users\SAMEEKSHA YS\aks-webapp> ls

Directory: C:\Users\SAMEEKSHA YS\aks-webapp

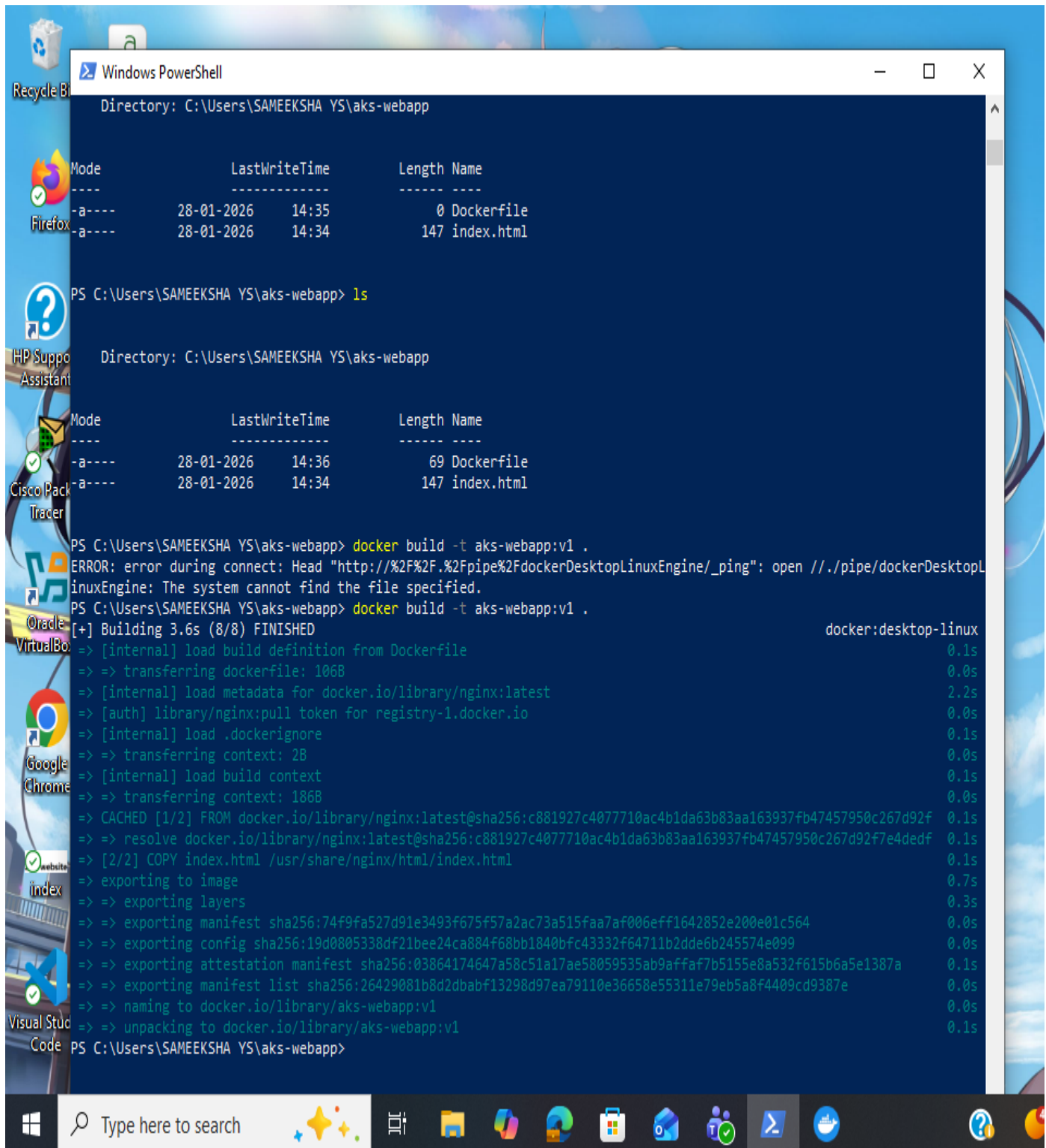
Mode                LastWriteTime         Length Name
----                -
-a-----         28-01-2026         14:35             0 Dockerfile
-a-----         28-01-2026         14:34          147 index.html
```

## Purpose

- Uses NGINX web server
  - Packages application into a container
  - Makes application portable
-

## STEP 6: Build Docker Image

**docker build -t aks-webapp:v1 .**



```
Windows PowerShell
Directory: C:\Users\SAMEEKSHA\YS\aks-webapp

Mode                LastWriteTime         Length Name
----                -
-a----             28-01-2026      14:35             0 Dockerfile
-a----             28-01-2026      14:34            147 index.html

PS C:\Users\SAMEEKSHA\YS\aks-webapp> ls

Directory: C:\Users\SAMEEKSHA\YS\aks-webapp

Mode                LastWriteTime         Length Name
----                -
-a----             28-01-2026      14:36             69 Dockerfile
-a----             28-01-2026      14:34            147 index.html

PS C:\Users\SAMEEKSHA\YS\aks-webapp> docker build -t aks-webapp:v1 .
ERROR: error during connect: Head "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file specified.
PS C:\Users\SAMEEKSHA\YS\aks-webapp> docker build -t aks-webapp:v1 .
[+] Building 3.6s (8/8) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 106B                             0.0s
=> [internal] load metadata for docker.io/library/nginx:latest  2.2s
=> [auth] library/nginx:pull token for registry-1.docker.io     0.0s
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                    0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 186B                                  0.0s
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f 0.1s
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf 0.1s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html      0.1s
=> exporting to image                                           0.7s
=> => exporting layers                                           0.3s
=> => exporting manifest sha256:74f9fa527d91e3493f675f57a2ac73a515faa7af006eff1642852e200e01c564 0.0s
=> => exporting config sha256:19d0805338df21bee24ca884f68bb1840bfc43332f64711b2dde6b245574e099 0.0s
=> => exporting attestation manifest sha256:03864174647a58c51a17ae58059535ab9affaf7b5155e8a532f615b6a5e1387a 0.1s
=> => exporting manifest list sha256:26429081b8d2dbabf13298d97ea79110e36658e55311e79eb5a8f4409cd9387e 0.0s
=> => naming to docker.io/library/aks-webapp:v1                 0.0s
=> => unpacking to docker.io/library/aks-webapp:v1              0.1s
PS C:\Users\SAMEEKSHA\YS\aks-webapp>
```

if start building without starting the docker the above error will come u need to start.

## Purpose

- Converts application into a Docker image
- Image can run consistently anywhere

---

## STEP 7: Push Image to Azure Container Registry

### Login to ACR

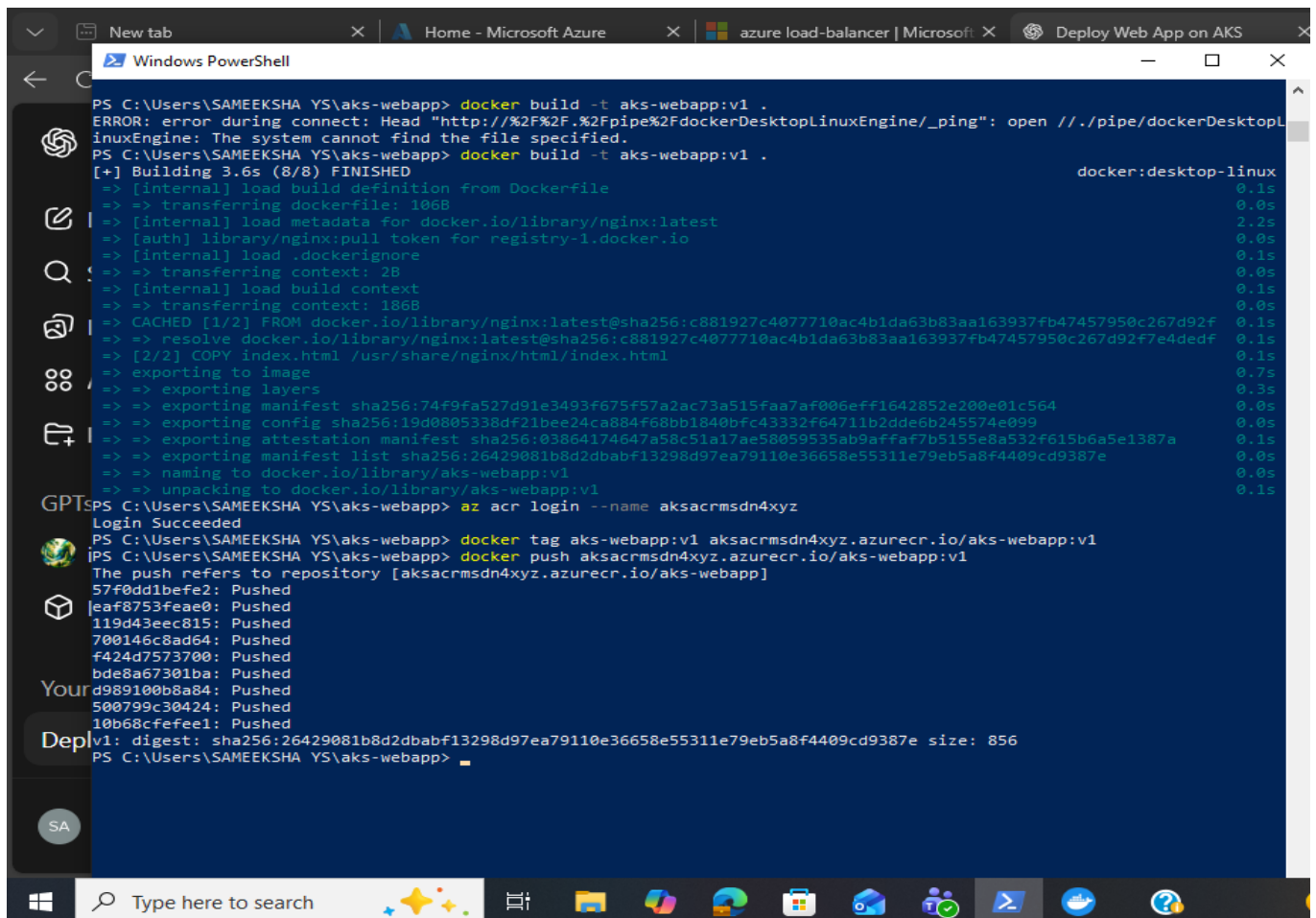
```
az acr login --name aksacrmsdn4xyz
```

### Tag Image

```
docker tag aks-webapp:v1 aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
```

### Push Image

```
docker push aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
```



```
PS C:\Users\SAMEEKSHA YS\aks-webapp> docker build -t aks-webapp:v1 .
ERROR: error during connect: Head "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file specified.
PS C:\Users\SAMEEKSHA YS\aks-webapp> docker build -t aks-webapp:v1 .
[+] Building 3.6s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                docker:desktop-linux
=> => transferring dockerfile: 106B                                              0.0s
=> [internal] load metadata for docker.io/library/nginx:latest                  2.2s
=> [auth] library/nginx:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                 0.1s
=> => transferring context: 2B                                                  0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 186B                                               0.0s
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f 0.1s
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf 0.1s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html                     0.1s
=> => exporting to image                                                         0.7s
=> => exporting layers                                                         0.3s
=> => exporting manifest sha256:74f9fa527d91e3493f675f57a2ac73a515faa7af006eff1642852e200e01c564      0.0s
=> => exporting config sha256:19d9805338df21bee24ca884f68bb1840bfc43332f64711b2dde6b245574e099      0.0s
=> => exporting attestation manifest sha256:03864174647a58c51a17ae58059535ab9affaf7b5155e8a532f615b6a5e1387a 0.1s
=> => exporting manifest list sha256:26429081b8d2dbabf13298d97ea79110e36658e55311e79eb5a8f4409cd9387e    0.0s
=> => naming to docker.io/library/aks-webapp:v1                               0.0s
=> => unpacking to docker.io/library/aks-webapp:v1                             0.1s
PS C:\Users\SAMEEKSHA YS\aks-webapp> az acr login --name aksacrmsdn4xyz
Login Succeeded
PS C:\Users\SAMEEKSHA YS\aks-webapp> docker tag aks-webapp:v1 aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
The push refers to repository [aksacrmsdn4xyz.azurecr.io/aks-webapp]
57f0dd1befe2: Pushed
1eaf8753feae0: Pushed
119d43eeca815: Pushed
700146c8ad64: Pushed
f424d7573700: Pushed
bde8a67301ba: Pushed
d989100b8a84: Pushed
500799c30424: Pushed
10b68cfeefe1: Pushed
v1: digest: sha256:26429081b8d2dbabf13298d97ea79110e36658e55311e79eb5a8f4409cd9387e size: 856
PS C:\Users\SAMEEKSHA YS\aks-webapp>
```



## **Purpose**

- **AKS cannot access local images**
  - **Image must exist in cloud registry**
  - **Enables secure image pulls**
- 

## **STEP 8: Create AKS Cluster (Azure Portal – Recommended)**

### **Portal Steps**

- **Azure Portal → Kubernetes services → Create**
- **Resource Group: aks-rg**
- **Cluster name: aks-cluster**
- **Node count: 2**
- **Authentication: Local accounts with Kubernetes RBAC**
- **Node security channel: Node Image**
- **Attach ACR: aksacrmsdn4xyz**

Microsoft Azure

Home > Kubernetes center (preview)

## Kubernetes center (preview) | Clusters

G7 CR Technologies India Pvt Ltd (g7cr.com)

Search resources, services, and docs (G+/)

Copilot

Overview

Fleets

**Clusters**

Managed namespaces

Search

+ Create Manage view Refresh Export to CSV Open query Group by none

Total clusters: 0 clusters 0 fleets • 0 managed namespaces [Browse quickstarts](#)

Effortlessly manage, scale, and perform operations on AKS or AKS Automatic clusters in one place for more streamlined and secure deployments. Easily create new clusters, add them to a Fleet, view usage and power states, and troubleshoot problems proactively. Support for ARC clusters will be available in a future release. [Learn more](#)

Filter for any field...

Subscription equals all Resource Group equals all Type equals all Location equals all

Fleet Manager equals all Power state equals all Managed namespaces equals

Kubernetes version equals all + Add filter

### No Kubernetes services to display

Showing 1 - 0 of 0. Display count: 200 [Give feedback](#)

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure

Home > Kubernetes center (preview) | Clusters

## Create Kubernetes cluster

Node security channel type

Security channel scheduler

Choose between local accounts or Microsoft authorization needs.

Authentication and Authorization

Node Image

- None**  
Your nodes won't have security updates applied automatically. You're responsible for all security updates.
- Unmanaged**  
OS updates are applied automatically through the OS built-in patching infrastructure.
- Security Patch**  
Security patches are applied, VHD is updated and existing machines are upgraded to that VHD. There may be disruptions when the security patches are applied. This option is only supported for Linux configurations.
- Node Image**  
AKS updates the nodes with a newly patched VHD containing security fixes and bug fixes on a weekly cadence.

Previous Next **Review + create** [Give feedback](#)

New tabCreate Kubernetes cluster - MicroDeploy Web App on AKSazure load-balancer | MicrosoftDeploy Web App on AKS

https://portal.azure.com/#create/Microsoft.AKS

Microsoft AzureSearch resources, services, and docs (G+)

Copilot

Home > Kubernetes center (preview) | Clusters

Create Kubernetes cluster

Azure Container Registry

Connect your cluster to an Azure Container Registry to enable seamless deployments from a private image registry.  
[Learn more](#)

Container registry

aksacrmsdn4xyz

Create new

Service mesh - Istio

Enable Istio to configure traffic management, maximize observability capabilities and reinforce service-to-service security measures without changing the application code. [Learn more](#)

Enable Istio☐

PreviousNextReview + create

Give feedback

Type here to search28°CENG15:0828-01-2026

New tabmicrosoft.aks-1769593338102Deploy Web App on AKSazure load-balancer | MicrosoftDeploy Web App on AKS

https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F94a3e059-5e41-48a4-b7e7-2abfabbd0b2a%2Fre...

Microsoft AzureSearch resources, services, and docs (G+)

Copilot

Home

microsoft.aks-1769593338102 | Overview

Deployment

Search

DeleteCancelRedeployDownloadRefresh

Overview

Inputs

Outputs

Template

Resource	Type	Status
ClusterOnboardingPut-e	Microsoft.Resources/d...	OK
CreatePromDCRA-20260	Microsoft.Resources/d...	OK
PrometheusAlerts-9a0a7	Microsoft.Resources/d...	OK
CreatePromDCR-202601	Microsoft.Resources/d...	OK
CreatePromRecordingRu	Microsoft.Resources/d...	OK
CreateAzureMonitorWor	Microsoft.Resources/d...	OK
ConnectAKStoACR-3944	Microsoft.Resources/d...	OK
aks-cluster	Microsoft.ContainerSe...	OK
InsightsMetricAlertsDepl	Microsoft.Resources/d...	OK

Cost Management

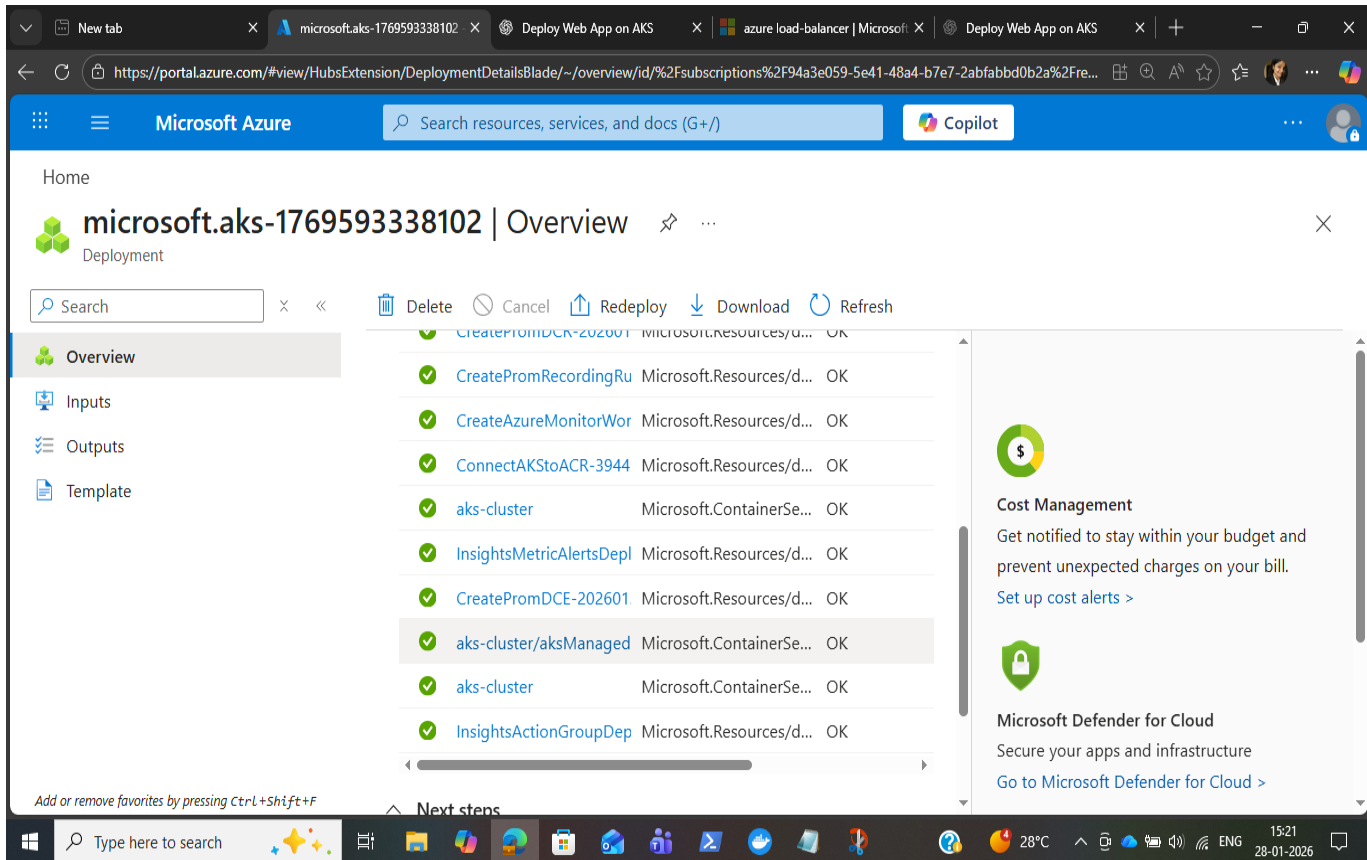
Get notified to stay within your budget and prevent unexpected charges on your bill.  
[Set up cost alerts >](#)

Microsoft Defender for Cloud

Secure your apps and infrastructure  
[Go to Microsoft Defender for Cloud >](#)

Add or remove favorites by pressing Ctrl+Shift+F

Type here to search28°CENG15:2128-01-2026



## Purpose

- **AKS provides managed Kubernetes**
- **Node Image ensures security patching**
- **Attaching ACR allows image pull permissions**

---

## STEP 9: Connect to AKS Cluster

**az aks get-credentials --resource-group aks-rg --name aks-cluster**

## Purpose

- **Downloads cluster credentials**
- **Configures kubectl to manage AKS**

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI

Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

```
trainingmsdnuser-4 [ ~ ]$ az aks get-credentials --resource-group aks-rg --name aks-cluster
```

```
trainingmsdnuser-4 [ ~ ]$ az aks get-credentials --resource-group aks-rg --name aks-cluster
```

-

Merged "aks-cluster" as current context in /home/trainingmsdnuser-4/.kube/config

## STEP 10: Verify Cluster

kubectl get nodes

```
trainingmsdnuser-4 [ ~ ]$ kubectl config get-contexts
```

CURRENT	NAME	CLUSTER	AUTHINFO	NAMESPACE
*	aks-cluster	aks-cluster	clusterUser_aks-rg_aks-cluster	

```
trainingmsdnuser-4 [ ~ ]$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-agentpool-28321478-vmss000000	Ready	<none>	11m	v1.33.6
aks-agentpool-28321478-vmss000001	Ready	<none>	11m	v1.33.6

```
trainingmsdnuser-4 [ ~ ]$
```



Type here to search



## **Purpose**

- **Confirms cluster is healthy**
  - **Nodes in Ready state**
- 

## **STEP 11: Create Kubernetes Deployment**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: aks-webapp-deployment**

**spec:**

**replicas: 2**

**selector:**

**matchLabels:**

**app: aks-webapp**

**template:**

**metadata:**

**labels:**

**app: aks-webapp**

**spec:**

**containers:**

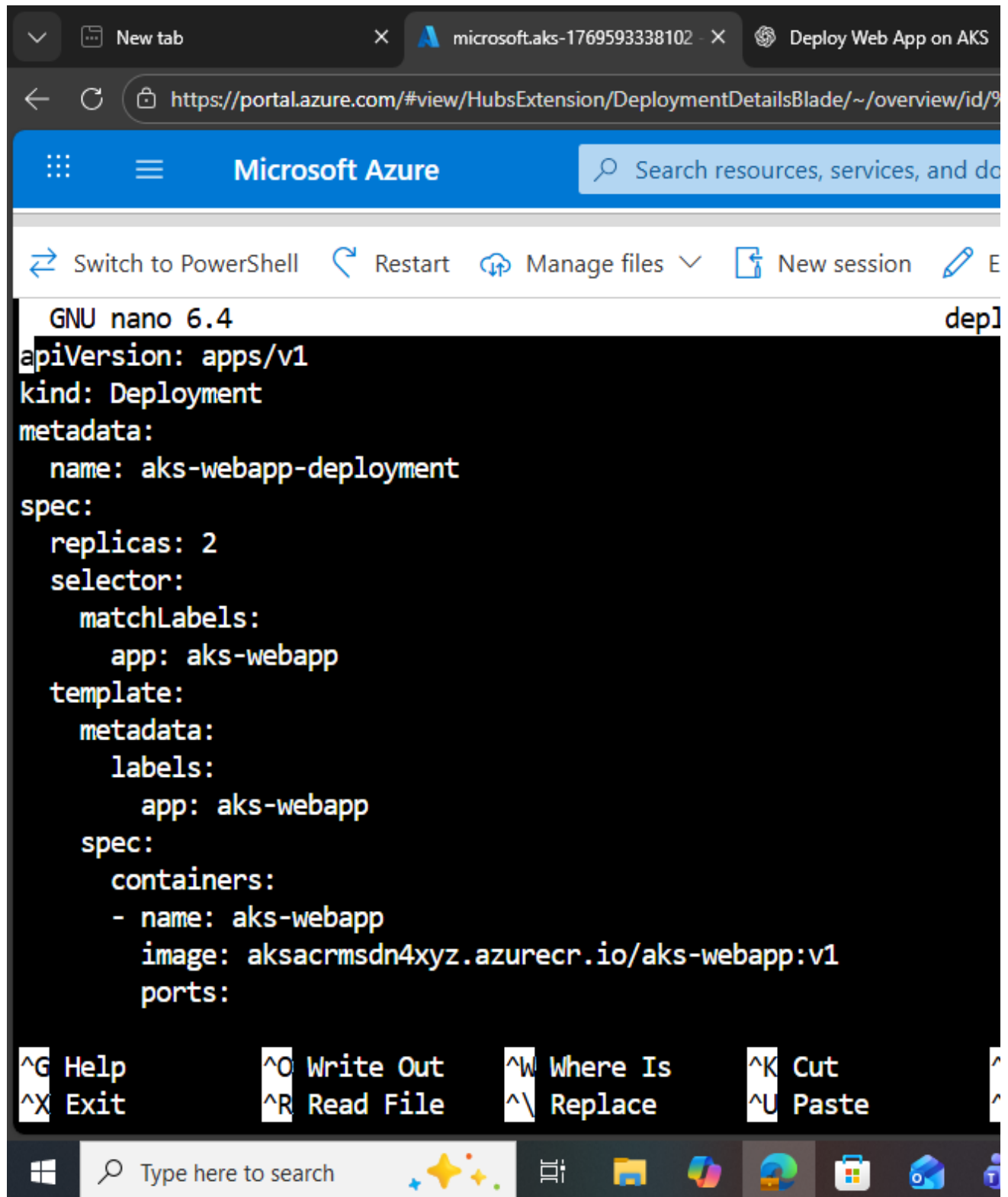
**- name: aks-webapp**

**image: aksacrmsdn4xyz.azurecr.io/aks-webapp:v1**

**ports:**

- containerPort: 80

kubectl apply -f deployment.yaml



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo and a search bar. Below the navigation bar, there's a toolbar with options like 'Switch to PowerShell', 'Restart', 'Manage files', 'New session', and 'Exit'. The main content area displays a terminal window with the GNU nano 6.4 editor. The terminal shows a Kubernetes deployment configuration for 'aks-webapp-deployment'. The configuration includes the API version, kind, metadata, spec (replicas, selector, template), and container details (name, image, ports). The terminal also shows a list of keyboard shortcuts at the bottom.

```
GNU nano 6.4 dep1
apiVersion: apps/v1
kind: Deployment
metadata:
  name: aks-webapp-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: aks-webapp
  template:
    metadata:
      labels:
        app: aks-webapp
    spec:
      containers:
      - name: aks-webapp
        image: aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
        ports:
```

Help Write Out Where Is Cut  
Exit Read File Replace Paste

Microsoft Azure portal interface showing a deployment configuration for an AKS web application.

**Browser Tabs:** New tab, microsoft.aks-1769593338102 - X, Deploy Web App on AKS

**URL:** <https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2F>

**Microsoft Azure Header:** Search resources, services, and docs

**Session Controls:** Switch to PowerShell, Restart, Manage files, New session, Edit

**Terminal Header:** GNU nano 6.4, deploy

```
metadata:
  name: aks-webapp-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: aks-webapp
  template:
    metadata:
      labels:
        app: aks-webapp
    spec:
      containers:
        - name: aks-webapp
          image: aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
          ports:
            - containerPort: 80
```

**Terminal Shortcuts:**

^G Help	^O Write Out	^W Where Is	^K Cut	^T
^X Exit	^R Read File	^_ Replace	^U Paste	^J

**Windows Taskbar:** Type here to search, File Explorer, Microsoft Store, Mail, Calendar, Photos, Settings, Task View, Search, Start



```
trainingmsdnuser-4 [ ~ ]$ nano deployment.yaml
trainingmsdnuser-4 [ ~ ]$ ls
deployment.yaml
trainingmsdnuser-4 [ ~ ]$ kubectl apply -f deployment.yaml
deployment.apps/aks-webapp-deployment created
trainingmsdnuser-4 [ ~ ]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
aks-webapp-deployment-644b49964f-nl99p	1/1	Running	0	16s
aks-webapp-deployment-644b49964f-rxwvf	1/1	Running	0	16s

```
trainingmsdnuser-4 [ ~ ]$
```

## Purpose

- Deployment manages pods
  - Ensures availability and self-healing
- 

## STEP 12: Expose Application Using Service

apiVersion: v1

kind: Service

metadata:

name: aks-webapp-service

spec:

type: LoadBalancer

selector:

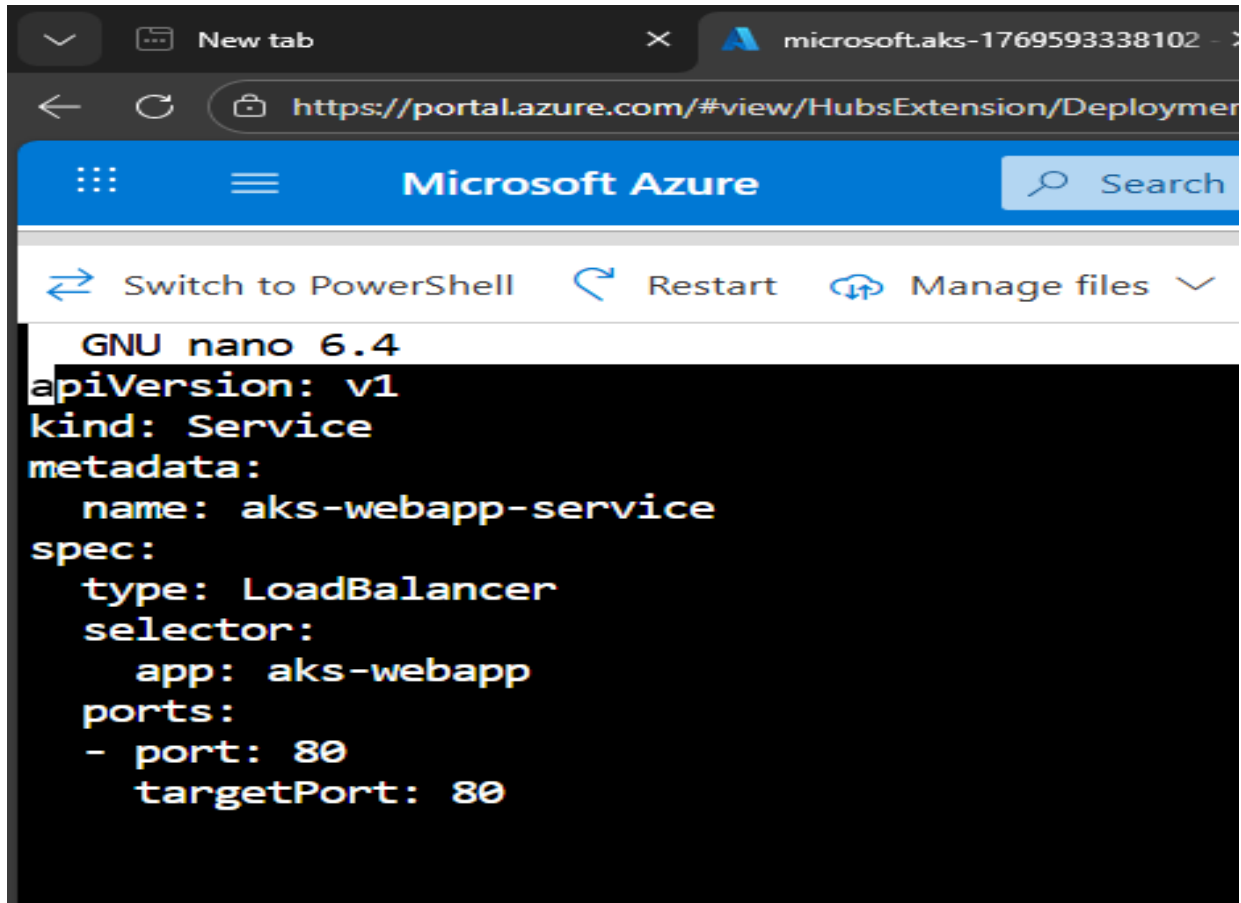
app: aks-webapp

ports:

- port: 80

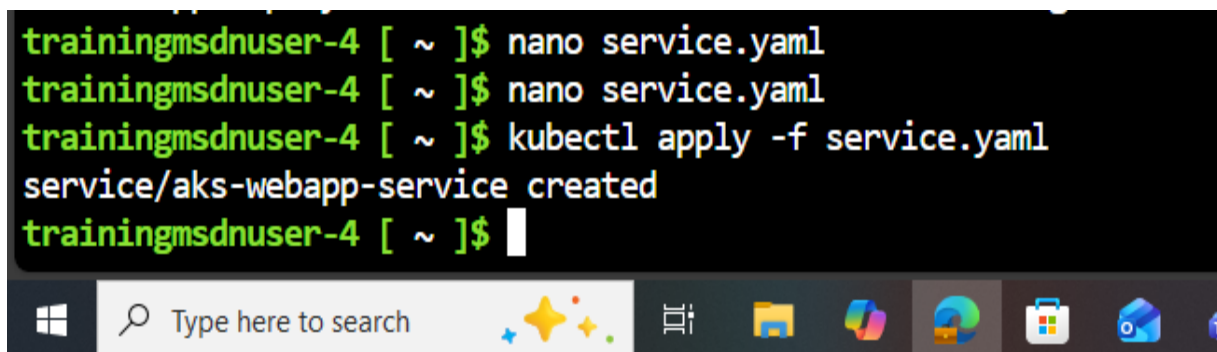
targetPort: 80

kubectl apply -f service.yaml



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the 'Microsoft Azure' logo and a search bar. Below the navigation bar, there's a toolbar with options: 'Switch to PowerShell', 'Restart', and 'Manage files'. The main content area displays a YAML configuration for a Kubernetes Service, edited in the 'nano' text editor. The configuration is as follows:

```
apiVersion: v1
kind: Service
metadata:
  name: aks-webapp-service
spec:
  type: LoadBalancer
  selector:
    app: aks-webapp
  ports:
    - port: 80
      targetPort: 80
```



The screenshot shows a terminal window with the following commands and output:

```
trainingmsdnuser-4 [ ~ ]$ nano service.yaml
trainingmsdnuser-4 [ ~ ]$ nano service.yaml
trainingmsdnuser-4 [ ~ ]$ kubectl apply -f service.yaml
service/aks-webapp-service created
trainingmsdnuser-4 [ ~ ]$
```

The terminal window is part of a Windows environment, as indicated by the taskbar at the bottom with the Start button and search bar.

## Purpose

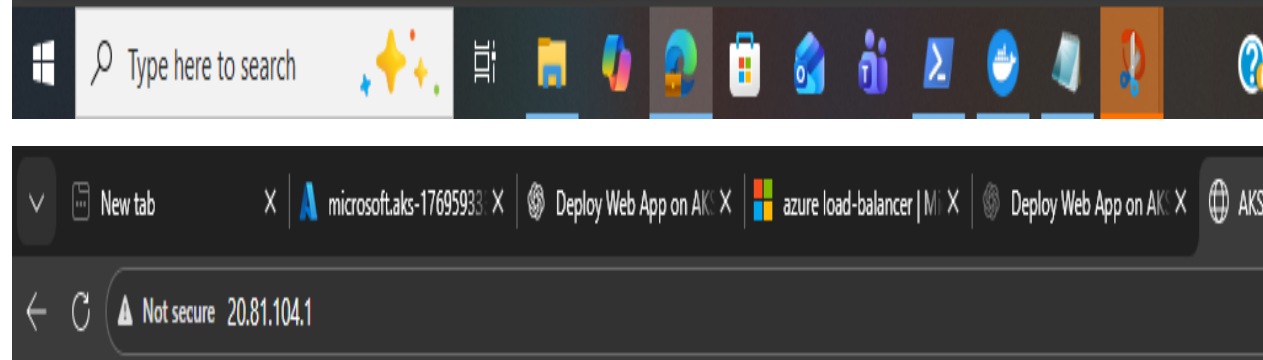
- Creates Azure Load Balancer
- Assigns public IP
- Routes traffic to pods

---

## STEP 13: Access Application

`kubectl get service aks-webapp-service`

```
trainingmsdnuser-4 [ ~ ]$ kubectl get service aks-webapp-service
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
aks-webapp-service  LoadBalancer  10.0.157.102   20.81.104.1    80:30911/TCP     19s
trainingmsdnuser-4 [ ~ ]$
```



The screenshot shows a Windows taskbar at the bottom with various application icons. Above it is a web browser window with multiple tabs. The active tab is titled '20.81.104.1' and shows a 'Not secure' warning. The main content area of the browser displays the text 'Hello from Azure Kubernetes Service!' in a large, bold, black serif font.

Hello from Azure Kubernetes Service!

Open browser:

<http://<EXTERNAL-IP>>

---

#### STEP 14: Cleanup (VERY IMPORTANT)

`az group delete --name aks-rg --yes --no-wait`

```
trainingmsdnuser-4 [ ~ ]$ kubectl get deployment
kubectl get service
kubectl get pods
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
aks-webapp-deployment	2/2	2	2	4m25s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
aks-webapp-service	LoadBalancer	10.0.157.102	20.81.104.1	80:30911/TCP	95s
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	22m

NAME	READY	STATUS	RESTARTS	AGE
aks-webapp-deployment-644b49964f-nl99p	1/1	Running	0	4m27s
aks-webapp-deployment-644b49964f-rxvwf	1/1	Running	0	4m27s

```
trainingmsdnuser-4 [ ~ ]$ az group delete --name aks-rg --yes --no-wait
```



## **Purpose**

- **Deletes all resources**
  - **Prevents unwanted charges**
- 

## **15. Key Learnings**

- **AKS architecture**
- **Docker + ACR integration**
- **Kubernetes Deployment & Service**
- **Load balancing and scalability**