

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

The screenshot shows a Microsoft Azure Cloud Shell session. The terminal window displays the following command and its output:

```
trainingmsdnuser-4 [ ~ ]$ az group create \
--name aks-rg \
--location eastus
{
  "id": "/subscriptions/94a3e059-5e41-48a4-b7e7-2abfabbd0b2a/resourceGroups/aks-rg",
  "location": "eastus",
  "managedBy": null,
  "name": "aks-rg",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

The Azure Cloud Shell interface includes a search bar, navigation links, and a Copilot button at the top. The taskbar at the bottom shows various pinned applications like File Explorer, Edge, and Copilot.

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
```

```
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/ak
ksacrmsdn4xyz",
  "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
```

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

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

I Directory: C:\Users\SAMEEKSHA\YS

Q 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
```

The screenshot shows a Windows terminal window with a dark blue background. It displays the command-line interface for creating a Dockerfile. The terminal shows the following sequence of commands:

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

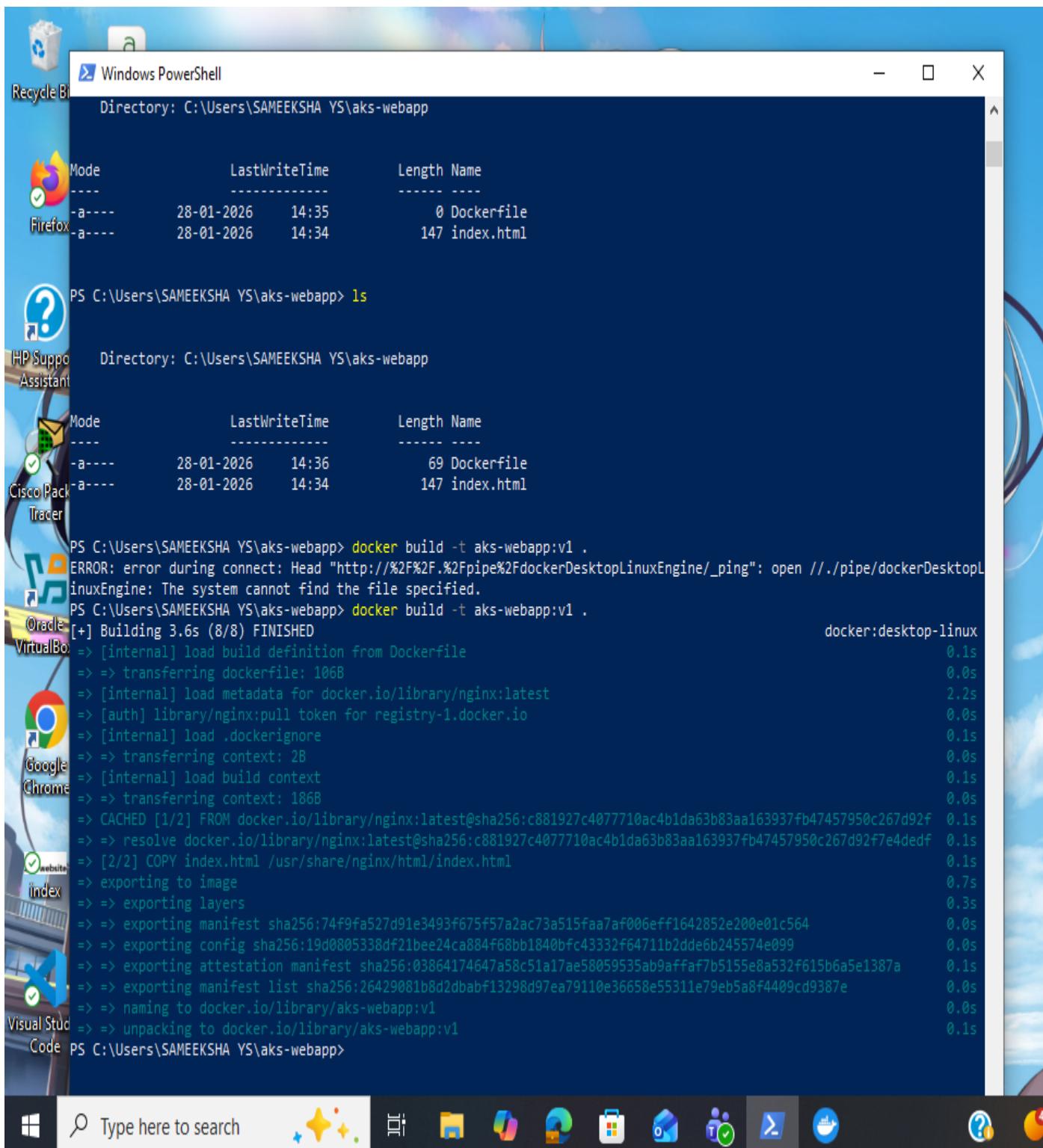
Below the terminal, a file explorer window is open, showing the directory structure. It lists two files: Dockerfile (0 bytes) and index.html (147 bytes). The file explorer interface includes icons for various file types like images, documents, and executables.

Purpose

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

STEP 6: Build Docker Image

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



A screenshot of a Windows desktop environment. In the center is a Windows PowerShell window titled "Windows PowerShell". The directory shown is "C:\Users\SAMEEKSHA\YS\aks-webapp". The user runs the command "ls" which lists two files: "Dockerfile" and "index.html". Then they run "docker build -t aks-webapp:v1 ." The output shows the build process, starting with "[+] Building 3.6s (8/8) FINISHED" and then detailing each step from loading the Dockerfile to exporting layers and manifest. The PowerShell window has a blue header bar with icons for file, edit, and search. The taskbar at the bottom includes icons for File Explorer, Edge browser, File History, Task View, and Start.

```
PS C:\Users\SAMEEKSHA\YS\aks-webapp> ls
Mode                LastWriteTime     Length Name
----                -----          0   Dockerfile
-a---    28-01-2026 14:35           0   Dockerfile
-a---    28-01-2026 14:34        147 index.html

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
=> => transferring dockerfile: 106B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 186B
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf
=> [2/2] COPY index.html /usr/share/nginx/html/index.html
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:74f9fa527d91e3493f675f57a2ac73a515faa7af006eff1642852e200e01c564
=> => exporting config sha256:19d0805338df21bee24ca884f68bb1840bfc43332f64711b2dde6b245574e099
=> => exporting attestation manifest sha256:03864174647a58c51a17ae58059535ab9affaf7b5155e8a532f615b6a5e1387a
=> => exporting manifest list sha256:26429081b8d2dbabf13298d97ea79110e36658e55311e79eb5a8f4409cd9387e
=> => naming to docker.io/library/aks-webapp:v1
=> => unpacking to docker.io/library/aks-webapp:v1
Code 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

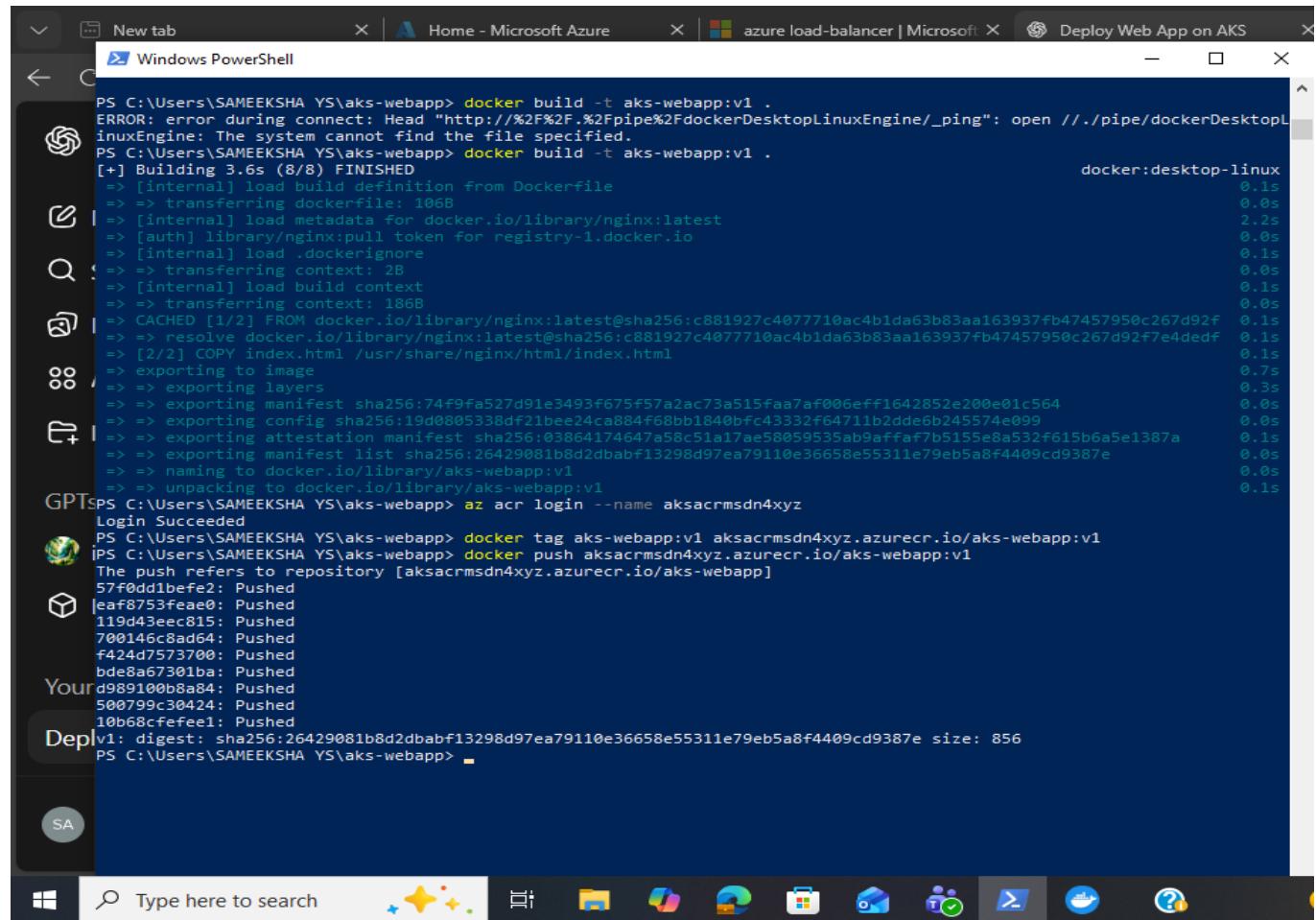
```
az acr login --name aksacrmsdn4xyz
```

Tag Image

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

Push Image

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



The screenshot shows a Windows PowerShell window with the following command history:

```
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
=> [internal] load metadata for docker.io/library/nginx:latest
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 186B
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf
=> [2/2] COPY index.html /usr/share/nginx/html/index.html
=> => exporting to image
=> => exporting layers
=> => exporting manifest sha256:74f9fa527d91e3493f675f57a2ac73a515faa7af006eff1642852e200e01c564
=> => exporting config sha256:19d0805338df21bee24ca884f68bb1840bfc4332f64711b2dde6b245574e099
=> => exporting attestation manifest sha256:03864174647a58c51a17ae58059535ab9affaf7b5155e8a532f615b6a5e1387a
=> => exporting manifest list sha256:26429081b8d2dbabf13298d97ea79110e36658e55311e79eb5a8f4409cd9387e
=> => naming to docker.io/library/aks-webapp:v1
=> => unpacking to docker.io/library/aks-webapp:v1
GPTSPS 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
PS C:\Users\SAMEEKSHA YS\aks-webapp> docker push aksacrmsdn4xyz.azurecr.io/aks-webapp:v1
The push refers to repository [aksacrmsdn4xyz.azurecr.io/aks-webapp]
57f0dd1bef2: Pushed
leaf8753feae0: Pushed
119d43ec815: Pushed
700146c8ad64: Pushed
f424d7573700: Pushed
bde8a67301ba: Pushed
d989100b8a84: Pushed
500799c30424: Pushed
10b68cfeef1: Pushed
Deploy 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

The screenshot shows the Microsoft Azure Kubernetes center (preview) Clusters page. The top navigation bar includes tabs for 'New tab', 'Kubernetes center (preview)', 'azure load-balancer | Microsoft', 'Deploy Web App on AKS', 'Deploy Web App on AKS', and a '+' button. The main title is 'Kubernetes center (preview) | Clusters'. A search bar and a Copilot button are at the top right. The left sidebar has links for 'Overview', 'Fleets', 'Clusters' (which is selected), and 'Managed namespaces'. The main content area displays a message: '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.' Below this is a 'Filter for any field...' input and several filter buttons: 'Subscription equals all', 'Resource Group equals all', 'Type equals all', 'Location equals all', 'Fleet Manager equals all', 'Power state equals all', 'Managed namespaces equals', and 'Kubernetes version equals all'. A 'No Kubernetes services to display' message is centered. At the bottom, it says 'Showing 1 - 0 of 0. Display count: 200' and a 'Give feedback' link.

The screenshot shows the Microsoft Azure Create Kubernetes cluster - Microsoft AKS page. The top navigation bar includes tabs for 'New tab', 'Create Kubernetes cluster - Microsoft AKS', 'azure load-balancer | Microsoft', 'Deploy Web App on AKS', 'Deploy Web App on AKS', and a '+' button. The main title is 'Create Kubernetes cluster'. The left sidebar has sections for 'Node security channel type' (with 'None' and 'Unmanaged' options), 'Choose between local accounts or Microsoft authorization needs', and 'Authentication and Authorization'. The right side shows a detailed view of the 'Node Image' section, which includes a dropdown menu, a 'None' option (describing manual security updates), an 'Unmanaged' option (describing OS patching via built-in infrastructure), and a 'Security Patch' option (describing weekly security updates). At the bottom, there are 'Previous' and 'Next' buttons, a 'Review + create' button, and a 'Give feedback' link.

New tab Create Kubernetes cluster - Microsoft Azure Deploy Web App on AKS azure load-balancer | Microsoft Deploy Web App on AKS

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

Microsoft Azure Search 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 [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

Previous Next Review + create Give feedback

Type here to search

28°C 15:08 28-01-2026 ENG

New tab microsoft.aks-1769593338102 - Microsoft Azure Deploy Web App on AKS azure load-balancer | Microsoft Deploy Web App on AKS

https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/-/overview/id/%2Fsubscriptions%2F94a3e059-5e41-48a4-b7e7-2abfabb0b2a%2Fre...

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

Copilot

Home

microsoft.aks-1769593338102 | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview

Inputs Outputs Template

Add or remove favorites by pressing **Ctrl+Shift+F**

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 >

Type here to search

28°C 15:21 28-01-2026 ENG

microsoft.aks-1769593338102 | Overview

Resource	Type	Status
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
CreatePromDCE-202601	Microsoft.Resources/d...	OK
aks-cluster/aksManaged	Microsoft.ContainerSe...	OK
aks-cluster	Microsoft.ContainerSe...	OK
InsightsActionGroupDep	Microsoft.Resources/d...	OK

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**

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 [ ~ ]$
```

Purpose

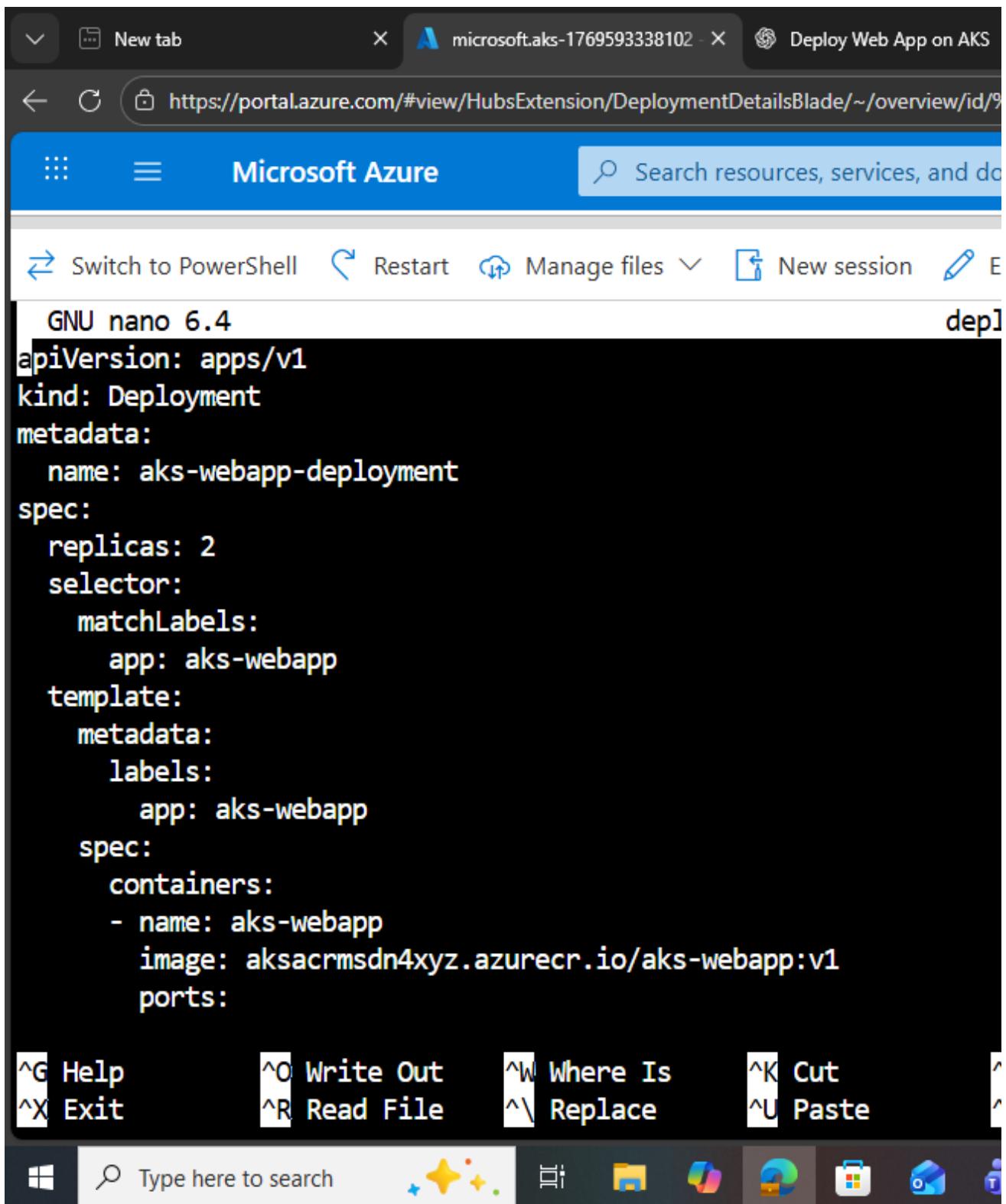
- Confirms cluster is healthy
 - Nodes in Ready state
-

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 a Microsoft Azure browser interface with a terminal window open. The terminal is running the GNU nano 6.4 editor and displays the following YAML configuration for a 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:
```

The terminal includes standard nano key bindings at the bottom:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste

The browser header shows tabs for "New tab" and "microsoft.aks-1769593338102" with a "Deploy Web App on AKS" button. The address bar shows the URL for the Azure portal.

New tab X microsoft.aks-1769593338102 Deploy Web App on AKS

← C https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fs

Microsoft Azure Search resources, services, and docs

Switch to PowerShell Restart Manage files New session Edit

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
```

Help Write Out Where Is Cut

Exit Read File Replace Paste

Type here to search

```
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-n199p  1/1     Running   0          16s
aks-webapp-deployment-644b49964f-rxvwf  1/1     Running   0          16s
trainingmsdnuser-4 [ ~ ]$
```



Purpose

- Deployment manages pods
- Ensures availability and self-healing

STEP 12: Expose Application Using Service

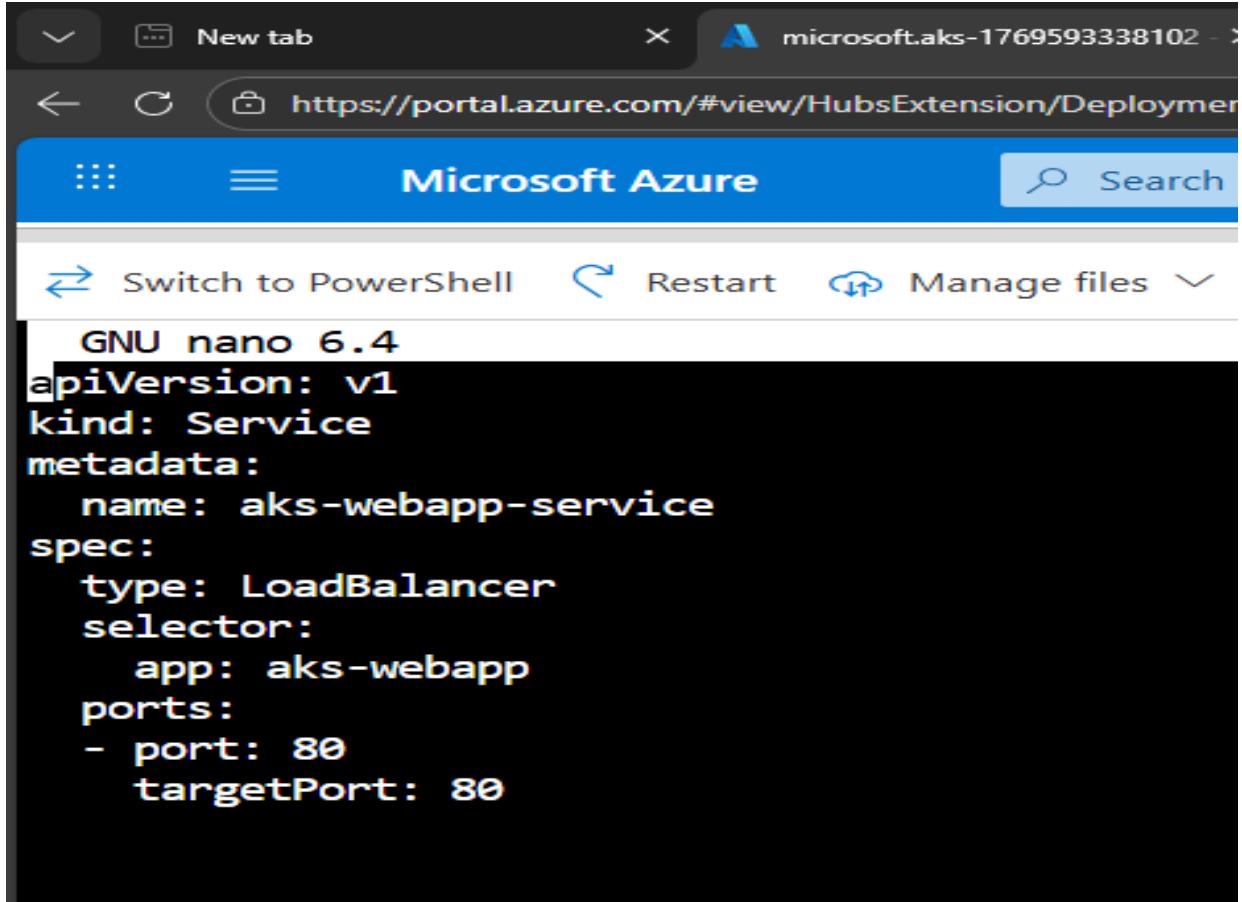
```
apiVersion: v1
kind: Service
metadata:
  name: aks-webapp-service
spec:
  type: LoadBalancer
  selector:
    app: aks-webapp
```

ports:

- port: 80

targetPort: 80

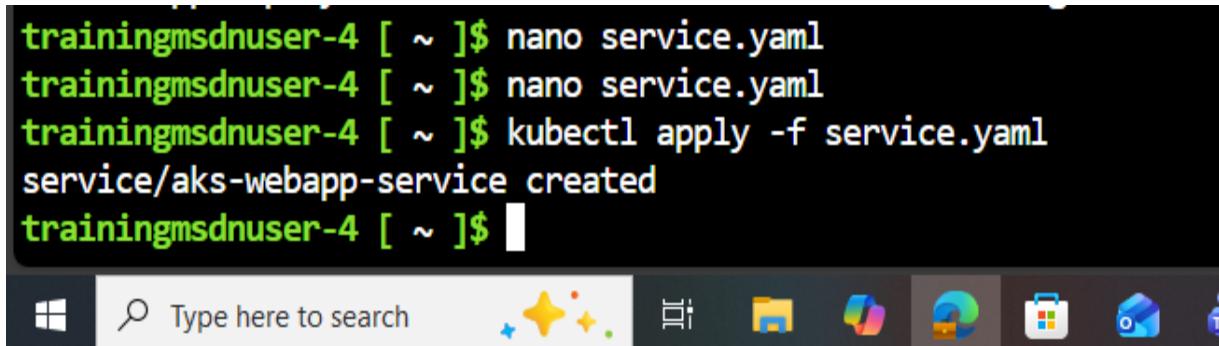
kubectl apply -f service.yaml



The screenshot shows the Microsoft Azure portal's deployment editor. At the top, there are tabs for 'New tab' and 'microsoft.aks-1769593338102'. Below the tabs, the URL is https://portal.azure.com/#view/HubsExtension/Deployment. The main area has a blue header with 'Microsoft Azure' and a search bar. Below the header, there are buttons for 'Switch to PowerShell', 'Restart', and 'Manage files'. The content area displays a YAML configuration for a service:

```
GNU nano 6.4
apiVersion: v1
kind: Service
metadata:
  name: aks-webapp-service
spec:
  type: LoadBalancer
  selector:
    app: aks-webapp
  ports:
  - port: 80
    targetPort: 80
```

```
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 screenshot shows a Windows terminal window with a black background and white text. The command history shows the user creating a service named 'aks-webapp-service' using the 'nano' editor to edit the 'service.yaml' file, then applying it with 'kubectl apply -f service.yaml'. The output indicates the service was created successfully.

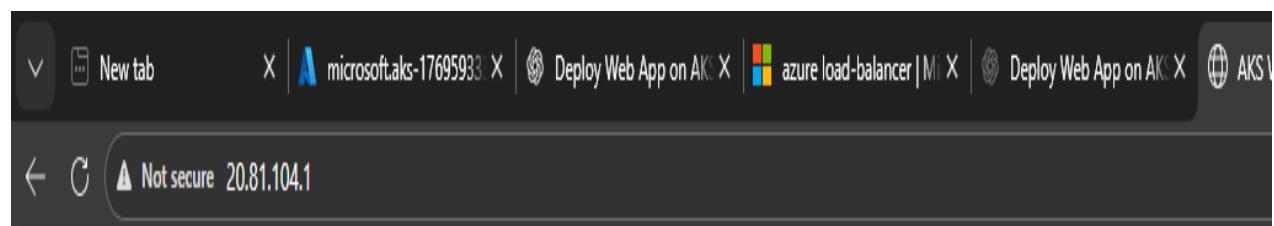
Purpose

- 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 [ ~ ]$
```



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