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

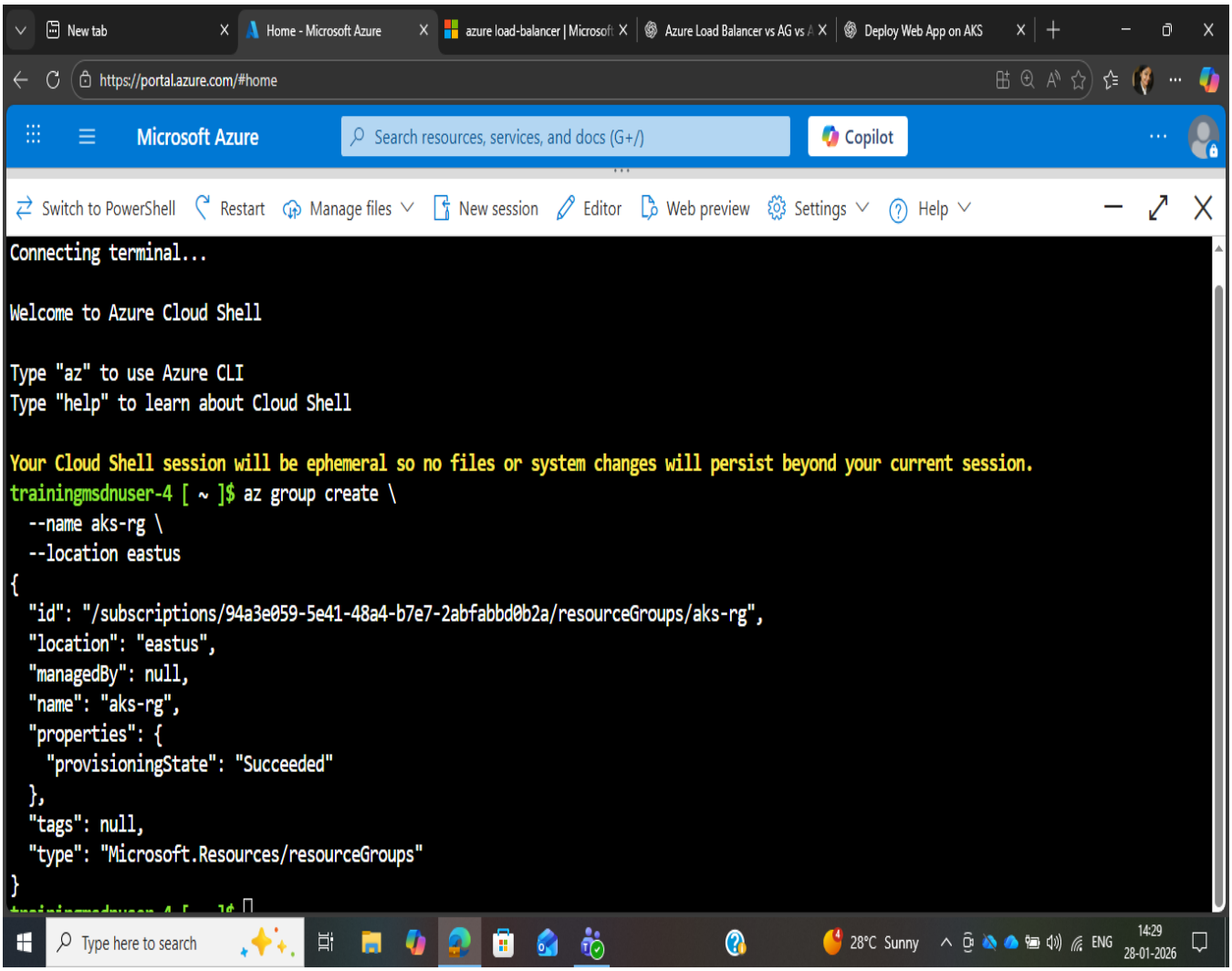
Purpose

- **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



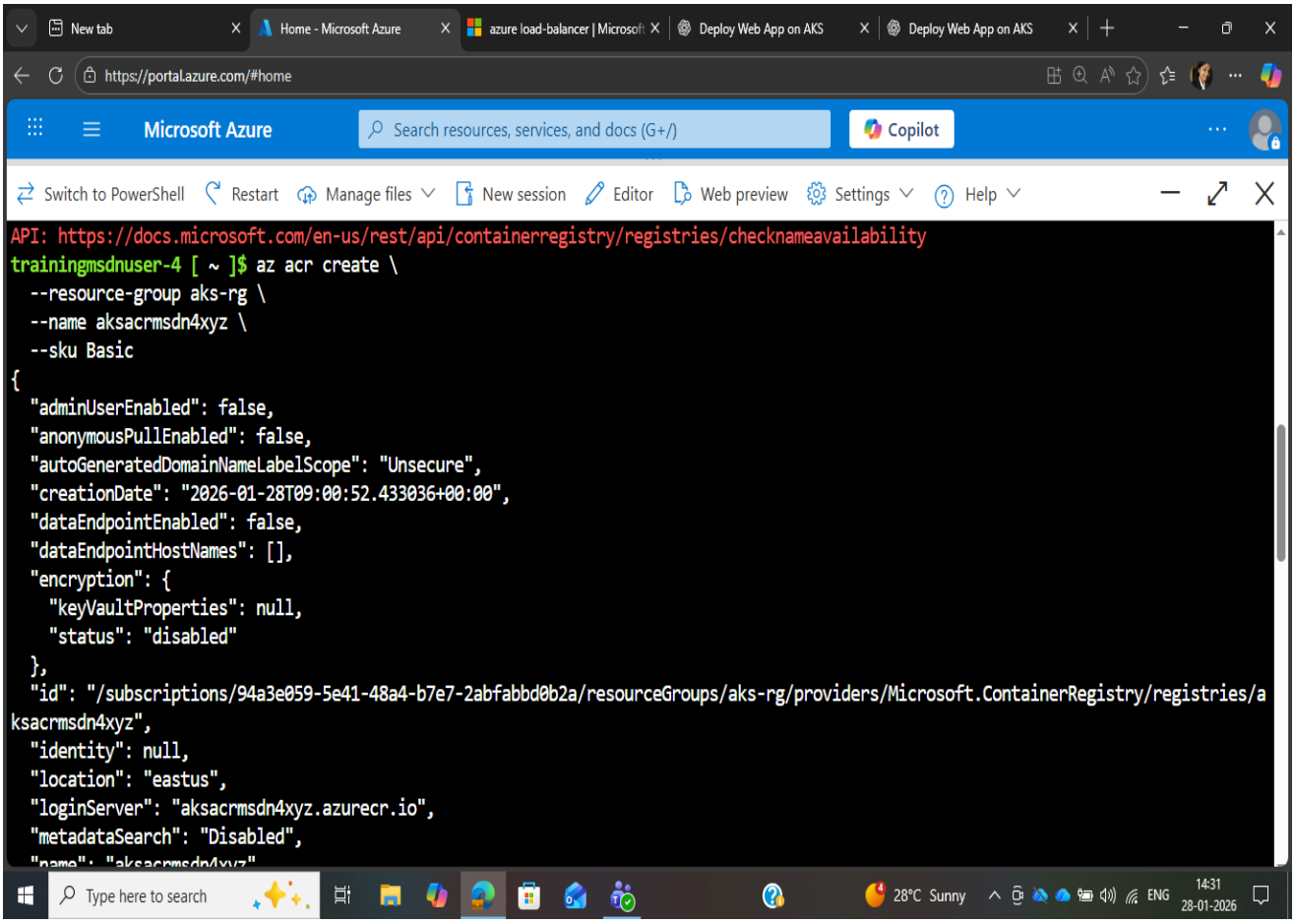
Purpose

- **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



Purpose

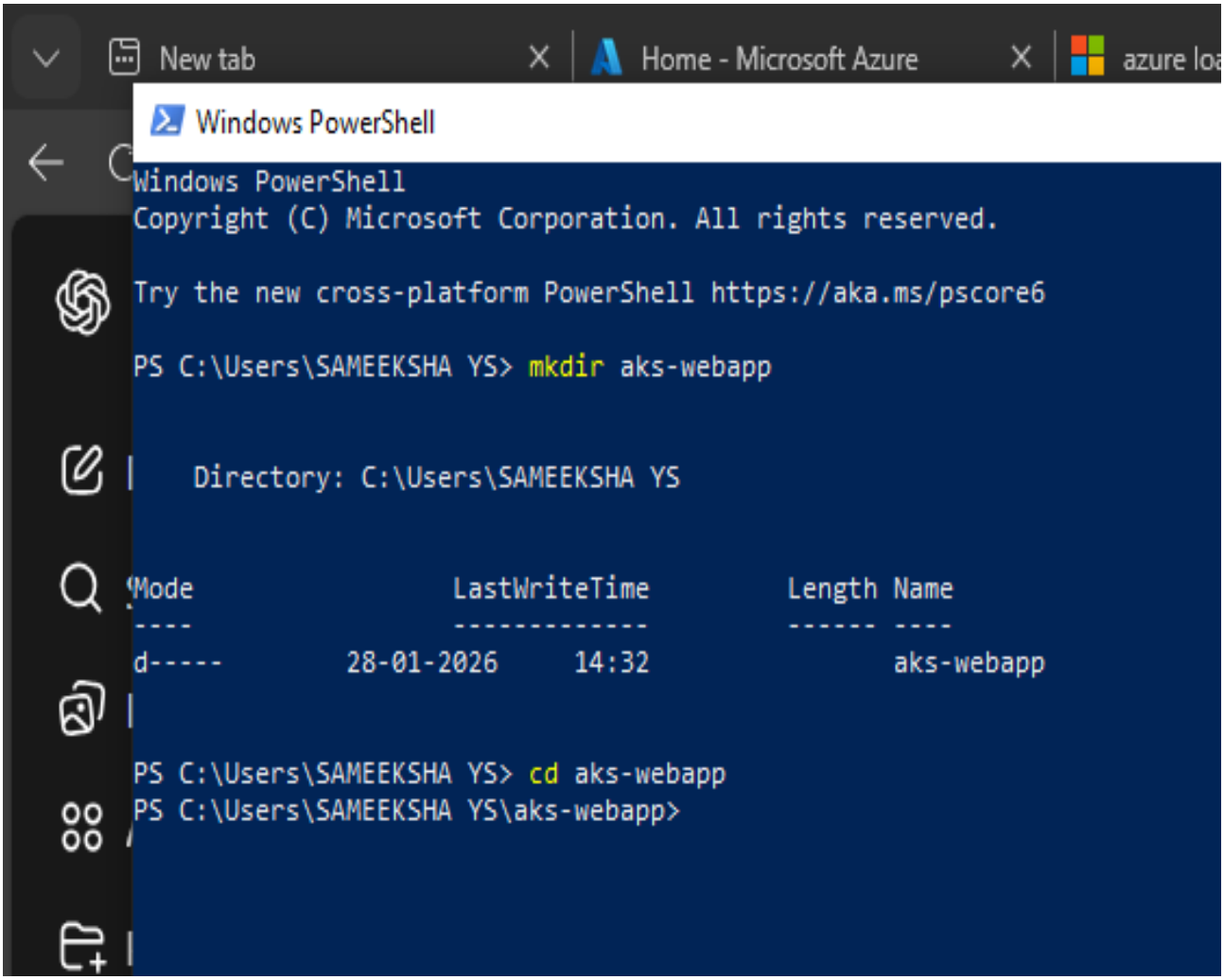
- **Stores Docker images privately**
- **More secure than public registries**
- **AKS pulls application images from ACR**

STEP 4: Build the Web Application Locally

Create Project Folder

mkdir aks-webapp

cd aks-webapp



Create index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>AKS Web App</title>
</head>
<body>
    <h1>Hello from Azure Kubernetes Service!</h1>
</body>
</html>
```

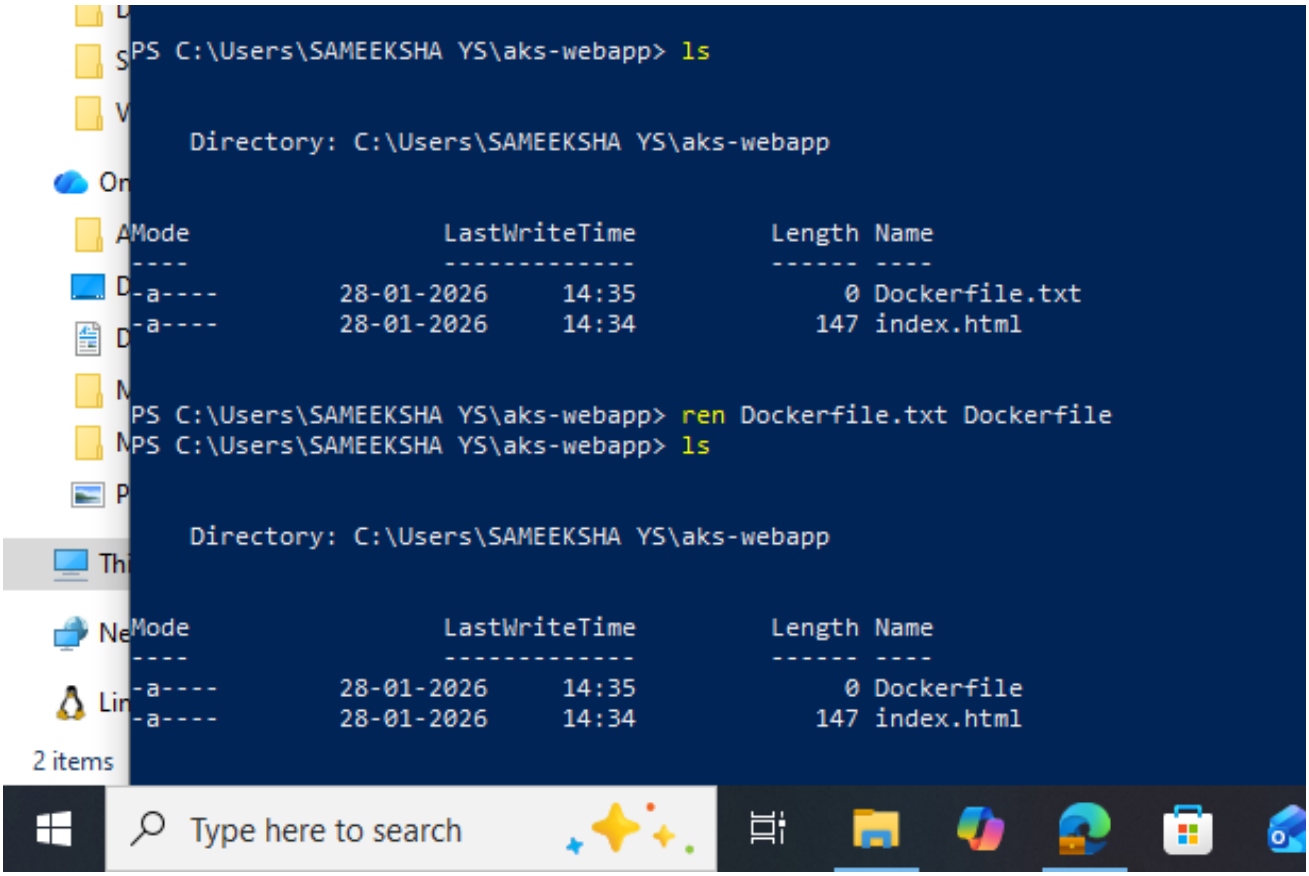
Purpose

- **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

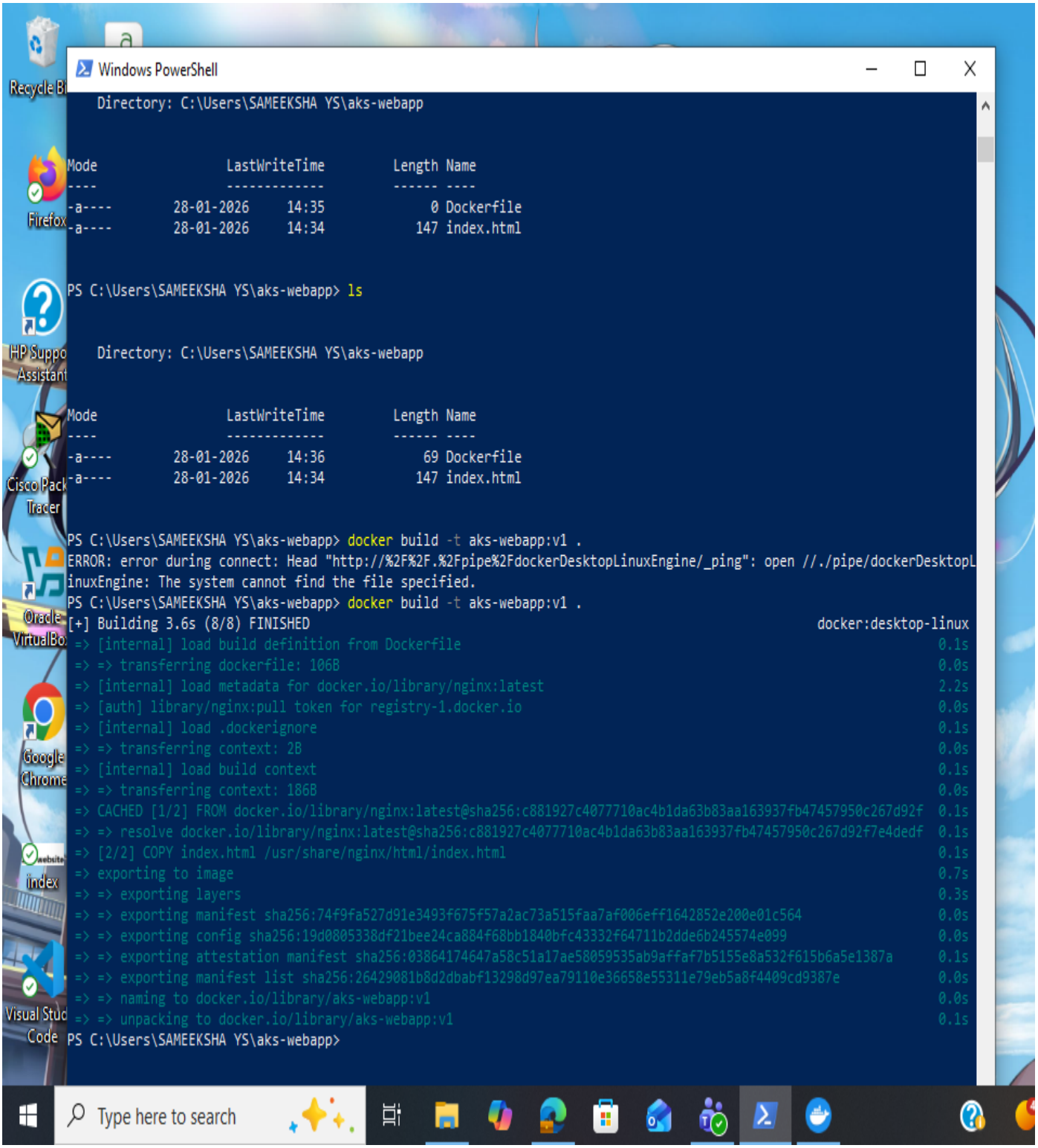


Purpose

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

STEP 6: Build Docker Image

docker build -t aks-webapp:v1 .



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

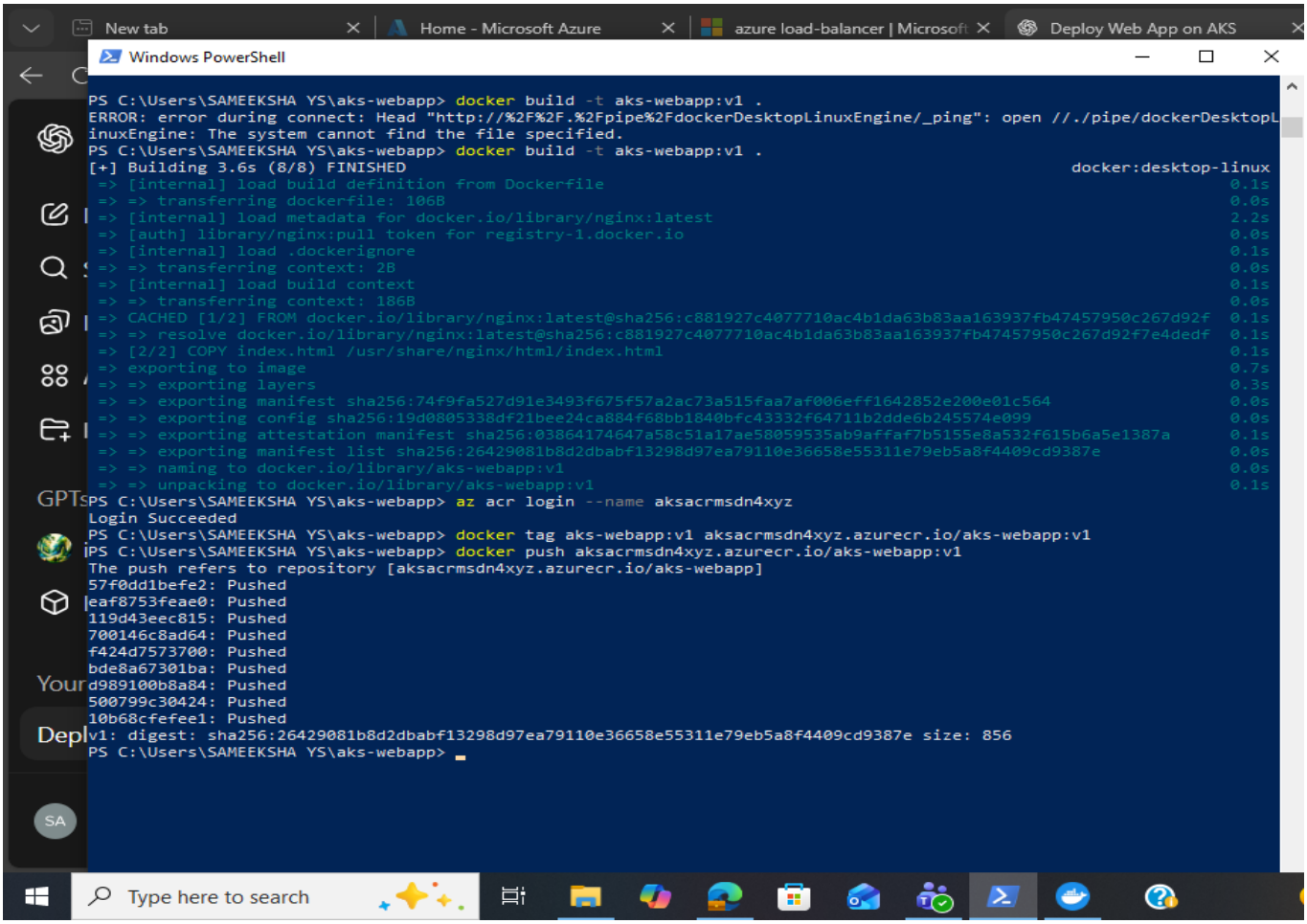
az acr login --name aksacrmsdn4xyz

Tag Image

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

Push Image

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



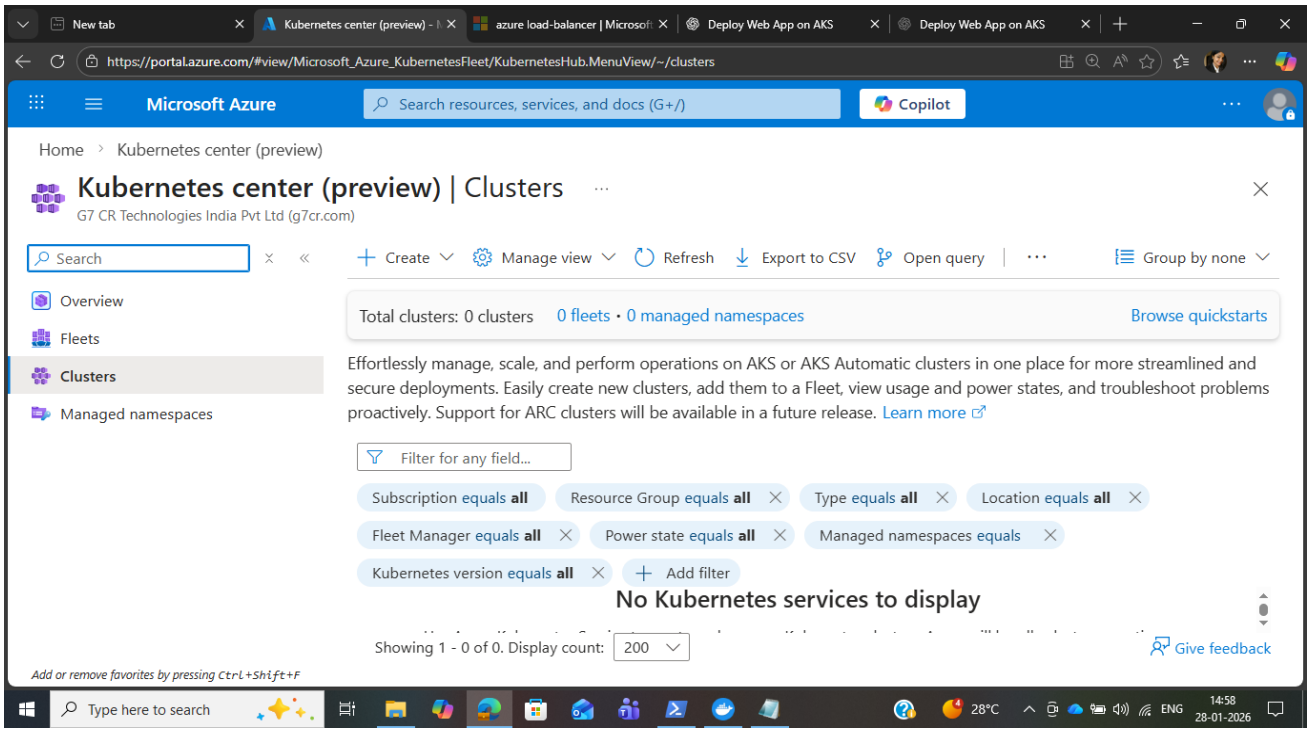
Purpose

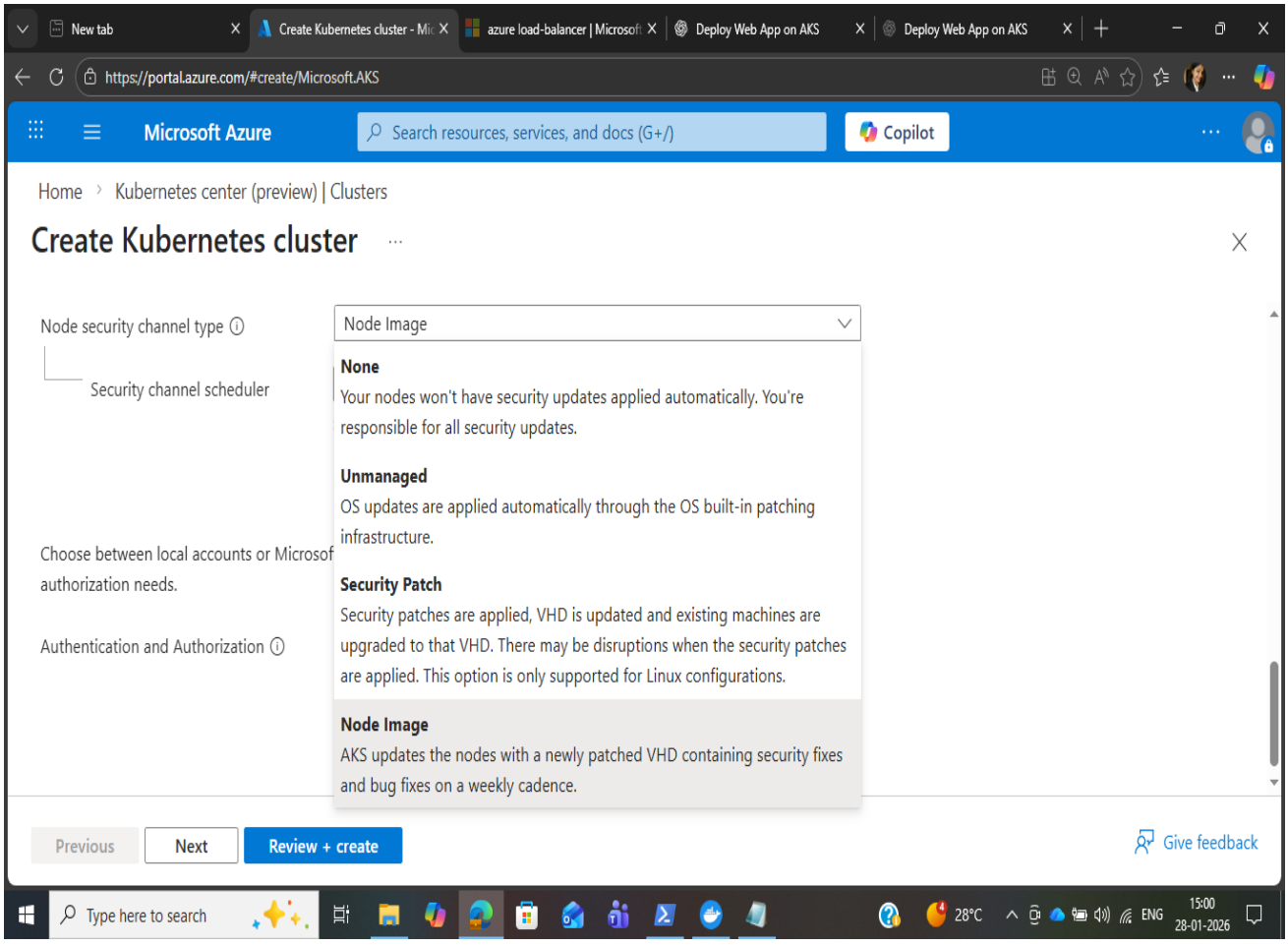
- **AKS cannot access local images**
- **Image must exist in cloud registry**
- **Enables secure image pulls**

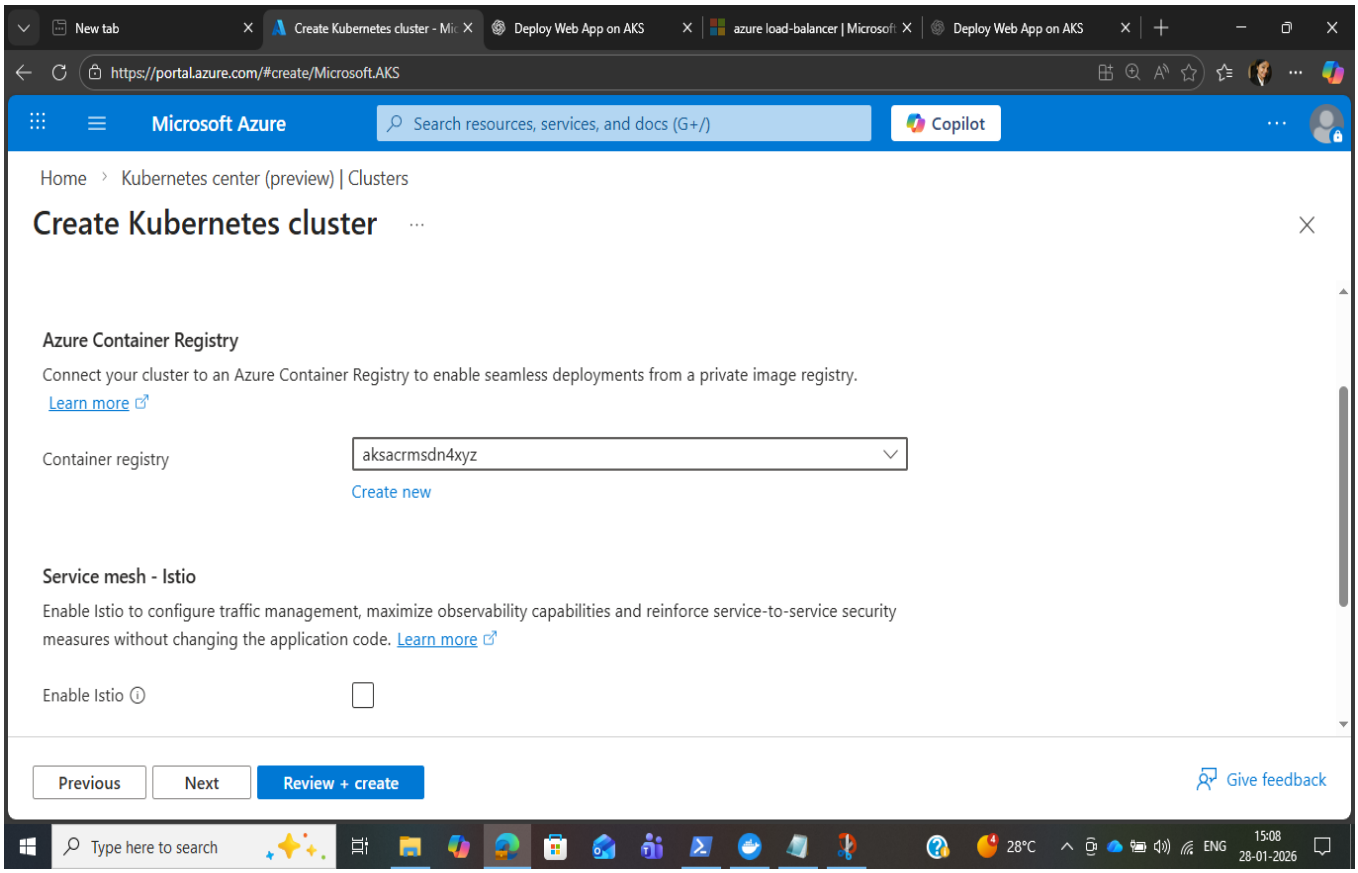
STEP 8: Create AKS Cluster (Azure Portal – Recommended)

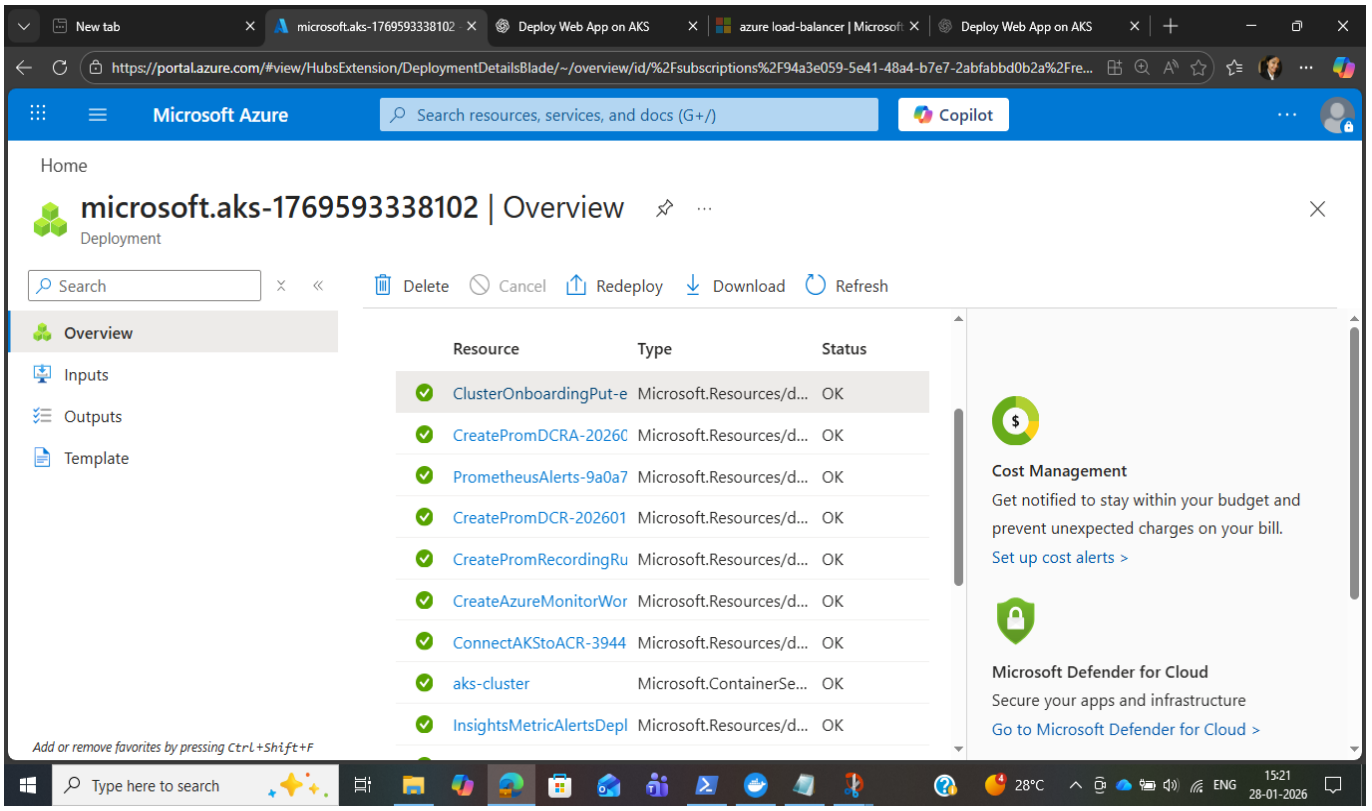
Portal Steps

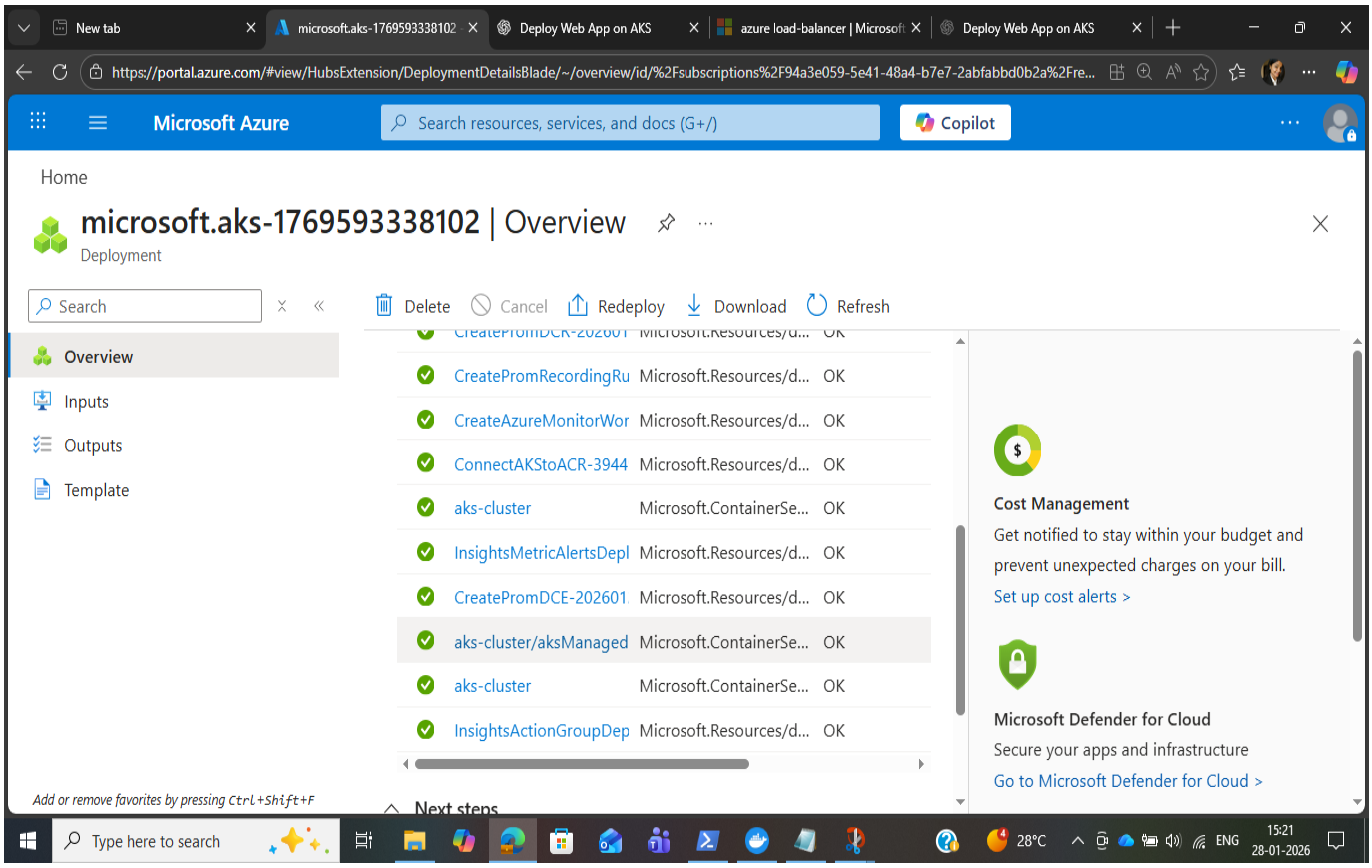
- **Azure Portal → Kubernetes services → Create**
- **Resource Group: aks-rg**
- **Cluster name: aks-cluster**
- **Node count: 2**
- **Authentication: Local accounts with Kubernetes RBAC**
- **Node security channel: Node Image**
- **Attach ACR: aksacrmsdn4xyz**


New tab
Kubernetes center (preview) - I
azure load-balancer | Microsoft
Deploy Web App on AKS
Deploy Web App on AKS
https://portal.azure.com/#view/Microsoft_Azure_KubernetesFleet/KubernetesHub.MenuView/~/clusters
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home
Kubernetes center (preview)
Kubernetes center (preview) | Clusters
G7 CR Technologies India Pvt Ltd (g7cr.com)
Search
Create
Manage view
Refresh
Export to CSV
Open query
Group by none
Overview
Fleets
Clusters
Managed namespaces
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
Type here to search
28°C
ENG
14:58
28-01-2026


New tab
Create Kubernetes cluster - Mic
azure load-balancer | Microsoft
Deploy Web App on AKS
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
Node security channel type
Security channel scheduler
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.
Choose between local accounts or Microsof
authorization needs.
Authentication and Authorization
Previous
Next
Review + create
Give feedback
Type here to search
28°C
ENG
15:00
28-01-2026


New tab
Create Kubernetes cluster - Mic
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
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
Previous
Next
Review + create
Give feedback
Type here to search
28°C
ENG
15:08
28-01-2026


New tab
microsoft.aks-1769593338102
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-b7e7-2abfabbd0b2a%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
Resource
Type
Status
ClusterOnboardingPut-e
Microsoft.Resources/d...
OK
CreatePromDCRA-2026(
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
Add or remove favorites by pressing Ctrl+Shift+F
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
ENG
15:21
28-01-2026



Purpose

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

STEP 9: Connect to AKS Cluster

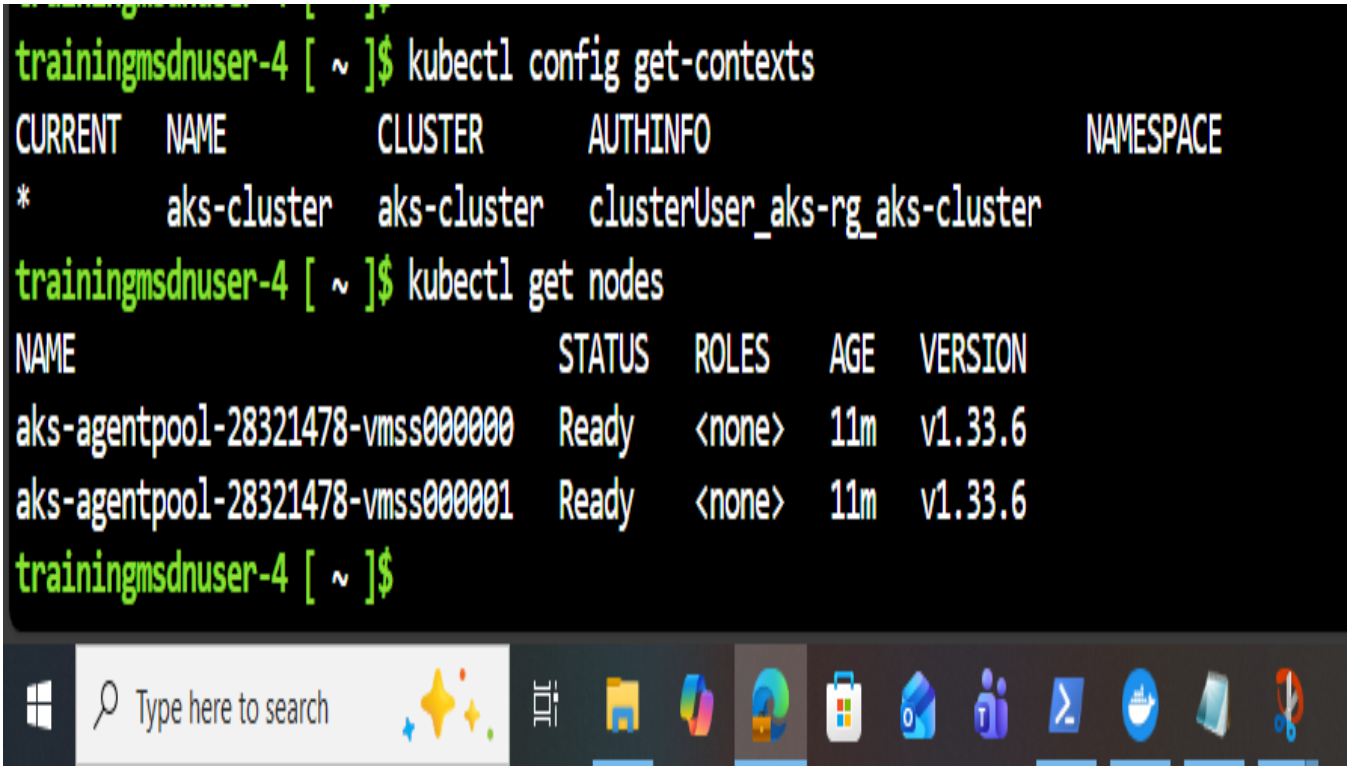
az aks get-credentials --resource-group aks-rg --name aks-cluster

Purpose

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

STEP 10: Verify Cluster

kubectl get nodes



Purpose

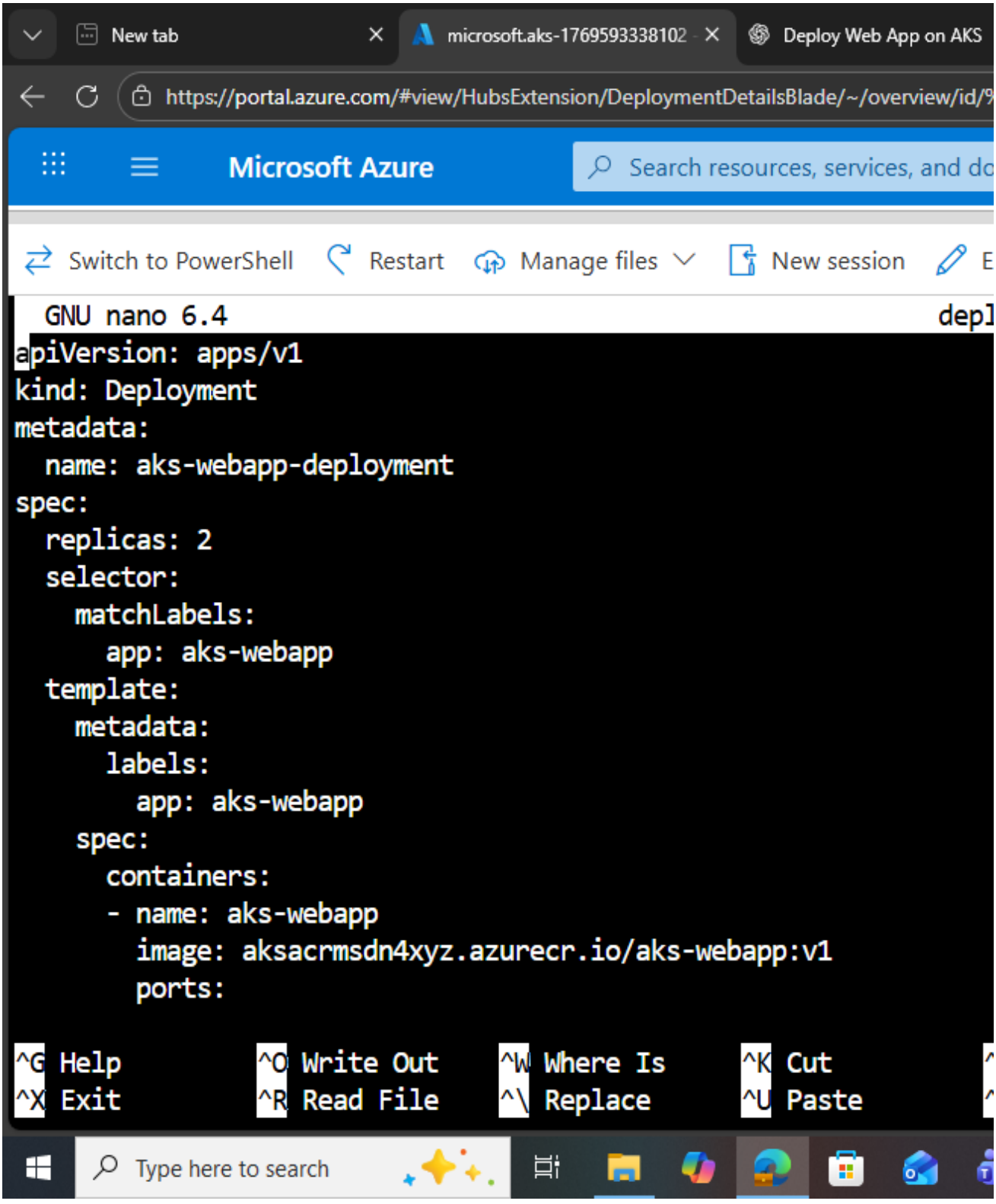
- **Confirms cluster is healthy**
- **Nodes in Ready state**

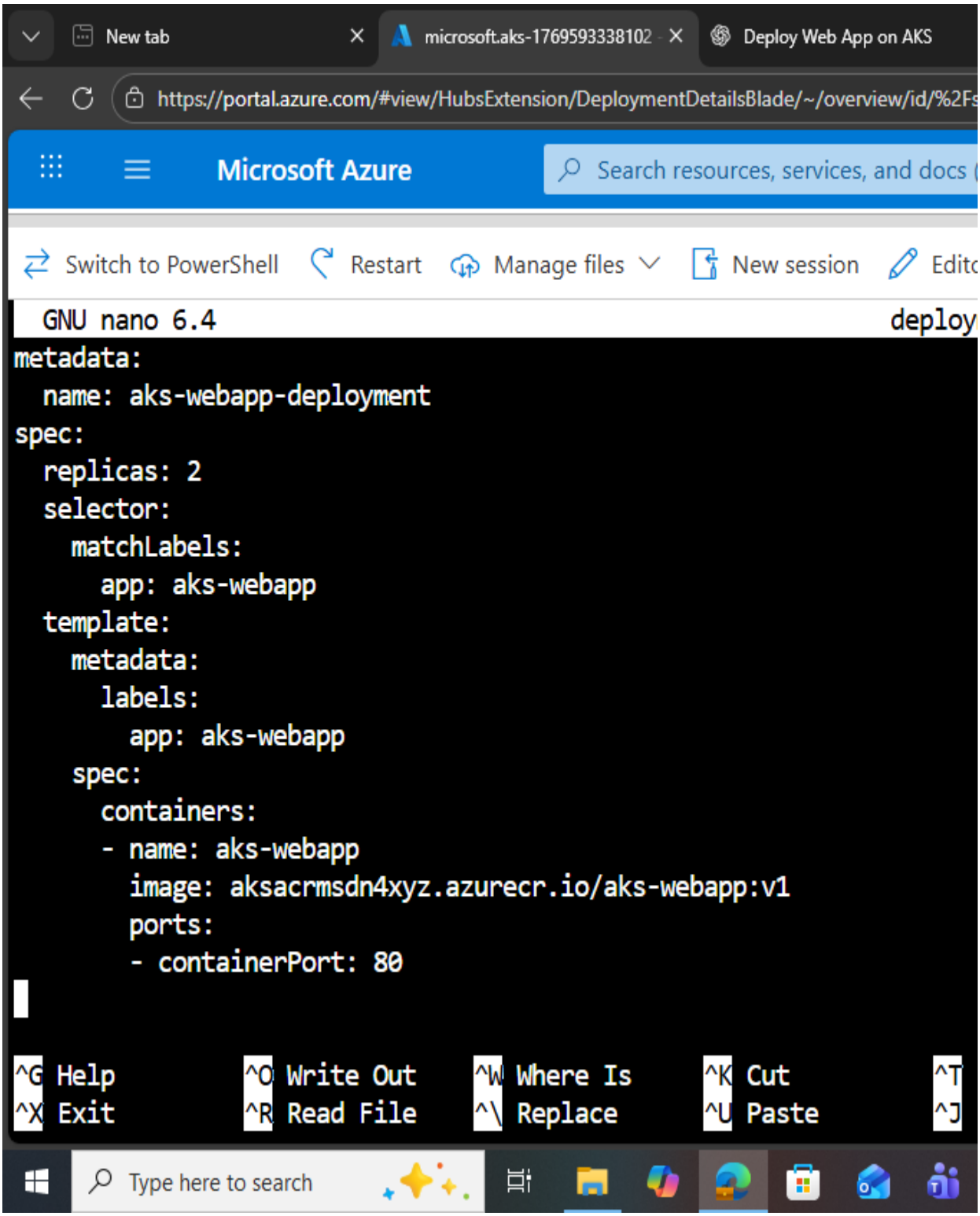
STEP 11: Create Kubernetes Deployment

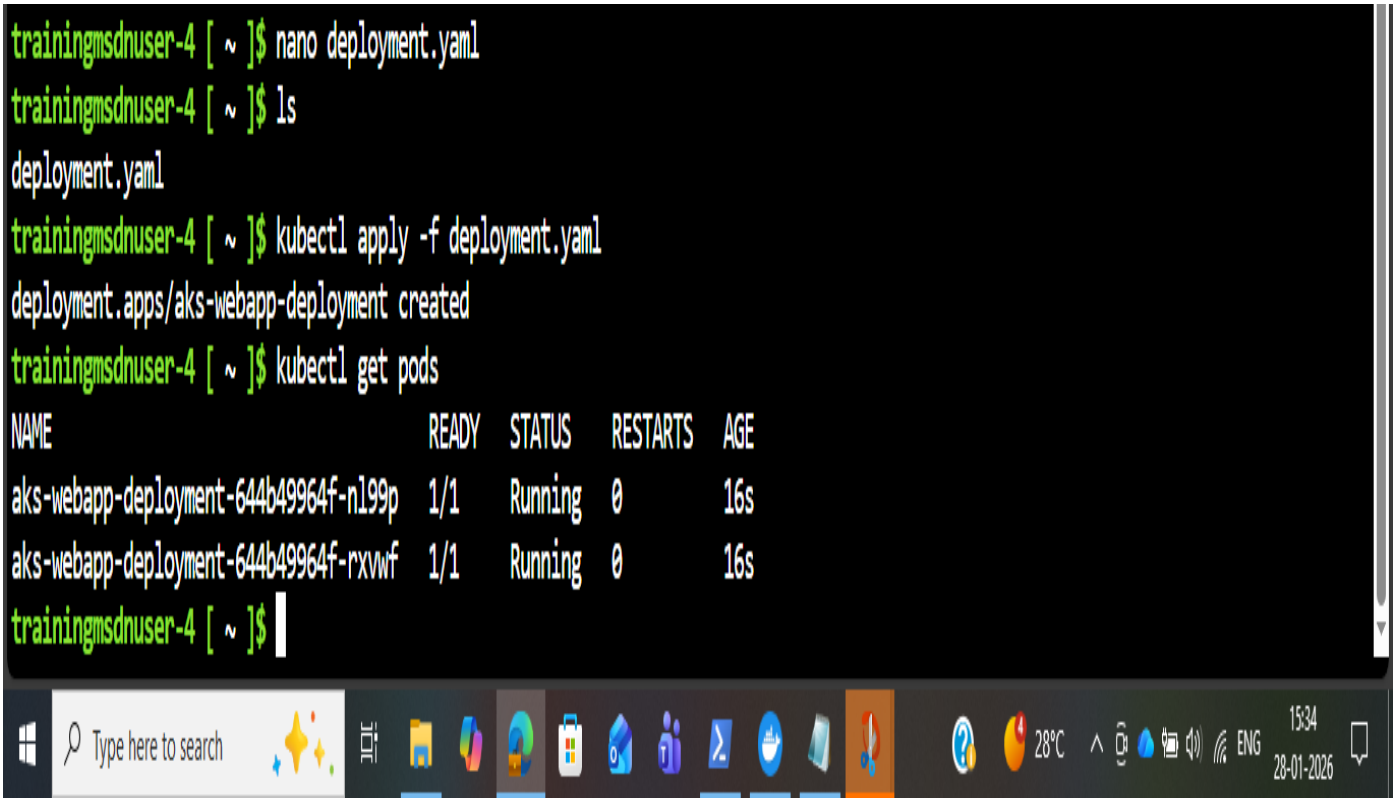
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: aks-webapp-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: aks-webapp
  template:
    metadata:
      labels:
        app: aks-webapp
    spec:
      containers:
      - name: aks-webapp
        image: aksacrmsdn4xyz.azurecr.io/aks-webapp:v1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

- containerPort: 80

kubectl apply -f deployment.yaml




New tab
microsoft.aks-1769593338102
Deploy Web App on AKS
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
Edito
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
^G Help
^O Write Out
^W Where Is
^K Cut
^T
^X Exit
^R Read File
^\ Replace
^U Paste
^J
Type here to search



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:

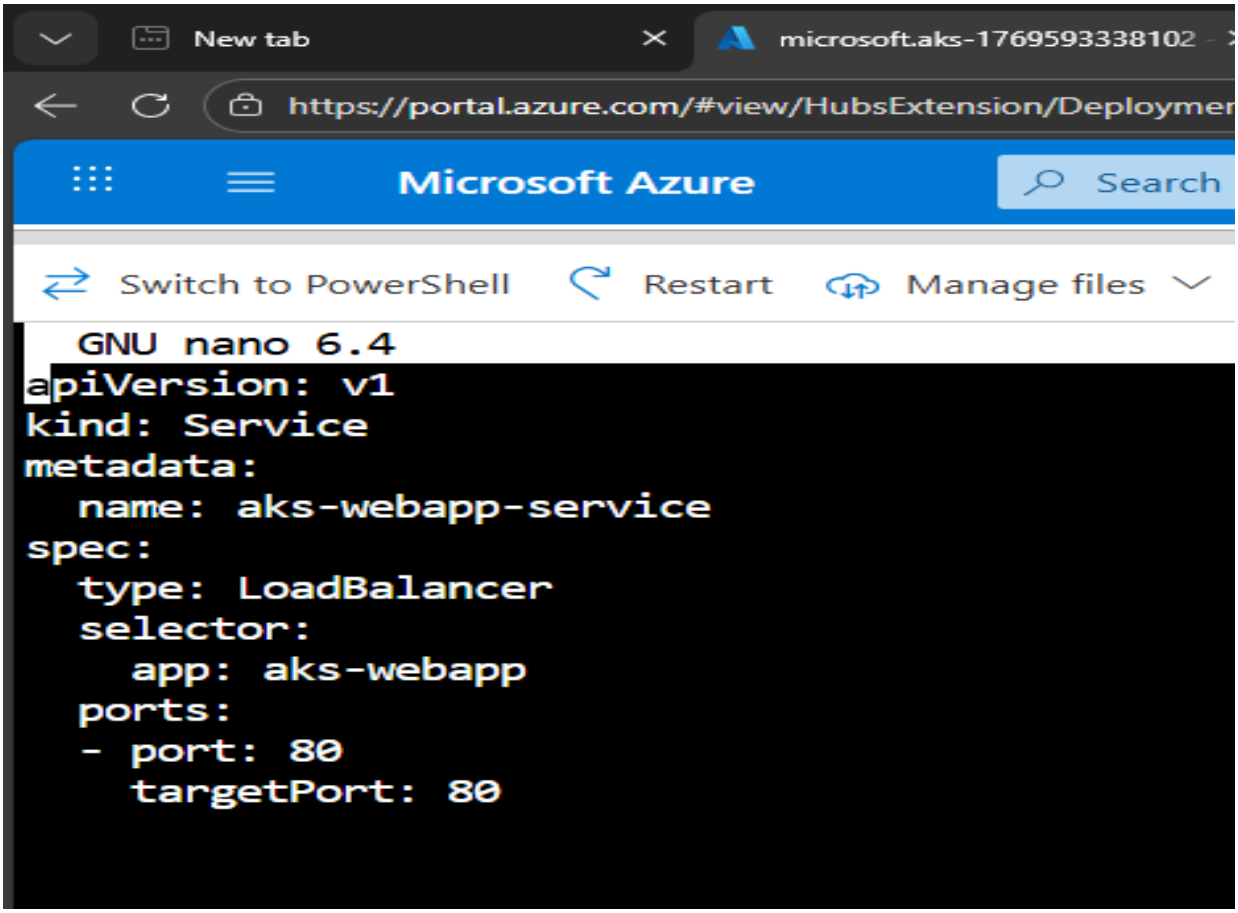
app: aks-webapp

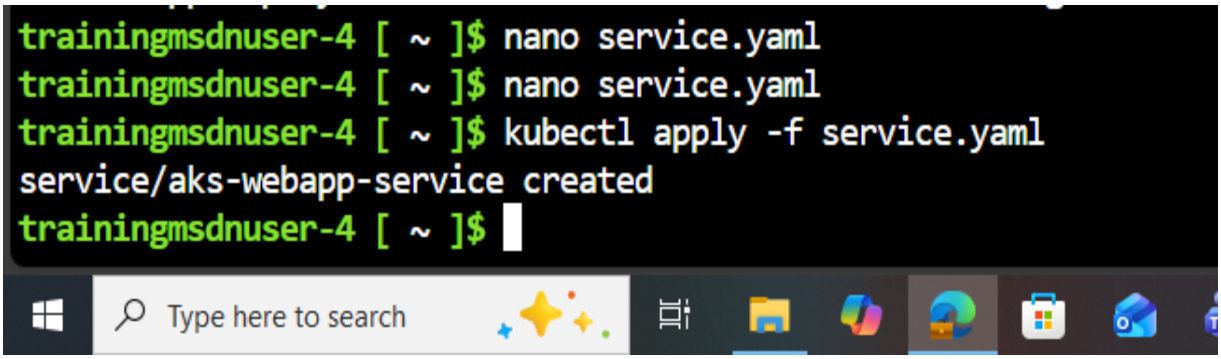
ports:

- port: 80

targetPort: 80

kubectl apply -f service.yaml



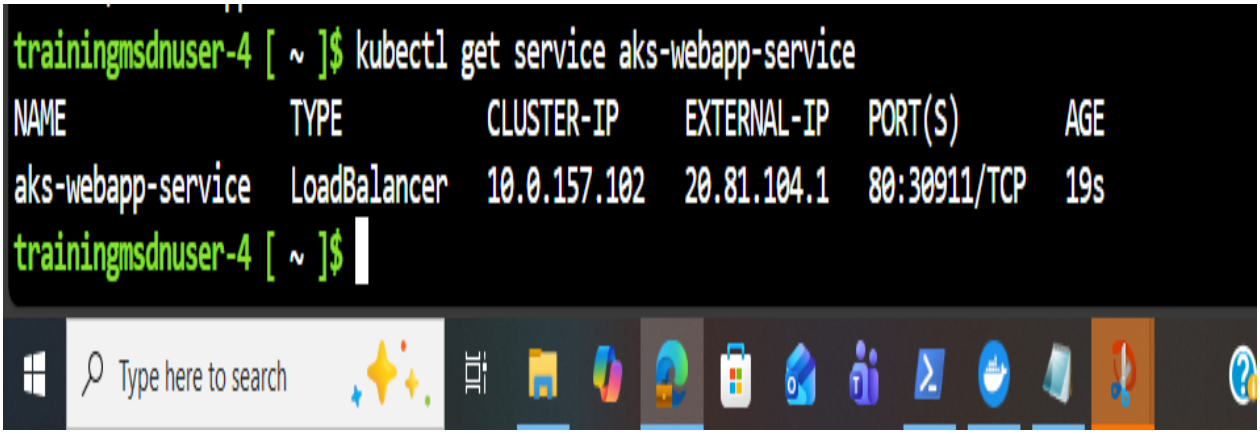


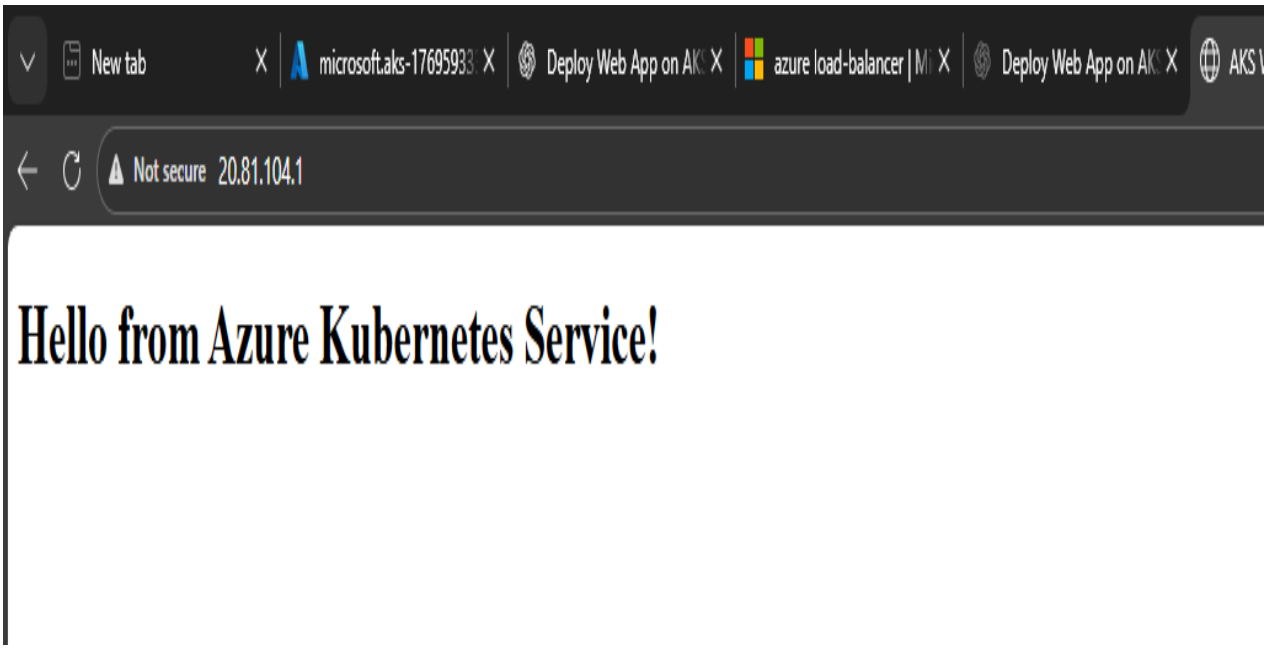
Purpose

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

STEP 13: Access Application

kubectl get service aks-webapp-service



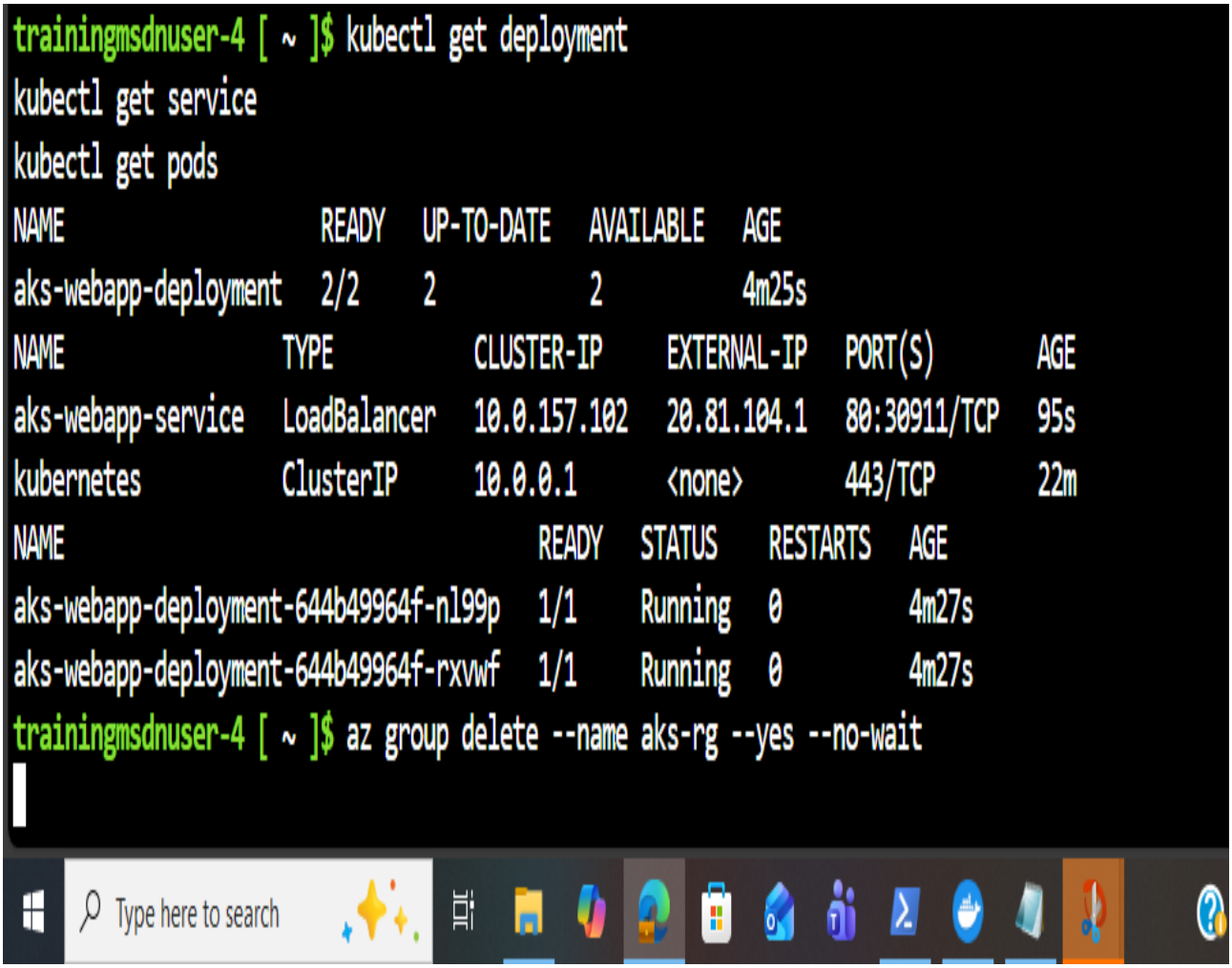


Open browser:

http://<EXTERNAL-IP>

STEP 14: Cleanup (VERY IMPORTANT)

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**