Deploy Application to Azure Kubernetes Service

In this Exercise we create asp.net application and Create image Locally using Docker Desktop Container installed in my Computer and Push the image to Docker Hub. The application is then deployed on AKS Cluster.

Basic Overview

To use Kubernetes Cluster, we can use cloud provider like AWS, Azure, GCP or any Infrastructure. Here we create Kubernetes Cluster using AKS (Azure Kubernetes service).

- To run our application in Kubernetes we create Docker image.
- Push the Image to Docker Hub or any Container.
- Alternatively Create manifest file to manage the scaling and Deployment of Containers.
- Automate using CI/CD Pipeline.
- Expose the Application by Creating Service like LB.
- After deploying your application in Kubernetes, manage and monitor application

Pre-Requisites:

Install Docker Desktop on your computer to containerize application.
Create Git Hub account.
Microsoft Visual Studio to Build asp.net application.
Microsoft Visual studio Code source code editor.
Microsoft Azure account to Create Azure Kubernetes Cluster

Create asp.net application on your computer.

Open PowerShell and Create source code folder. Type in the Command as below
Create new asp.net application with name **aksdemo**

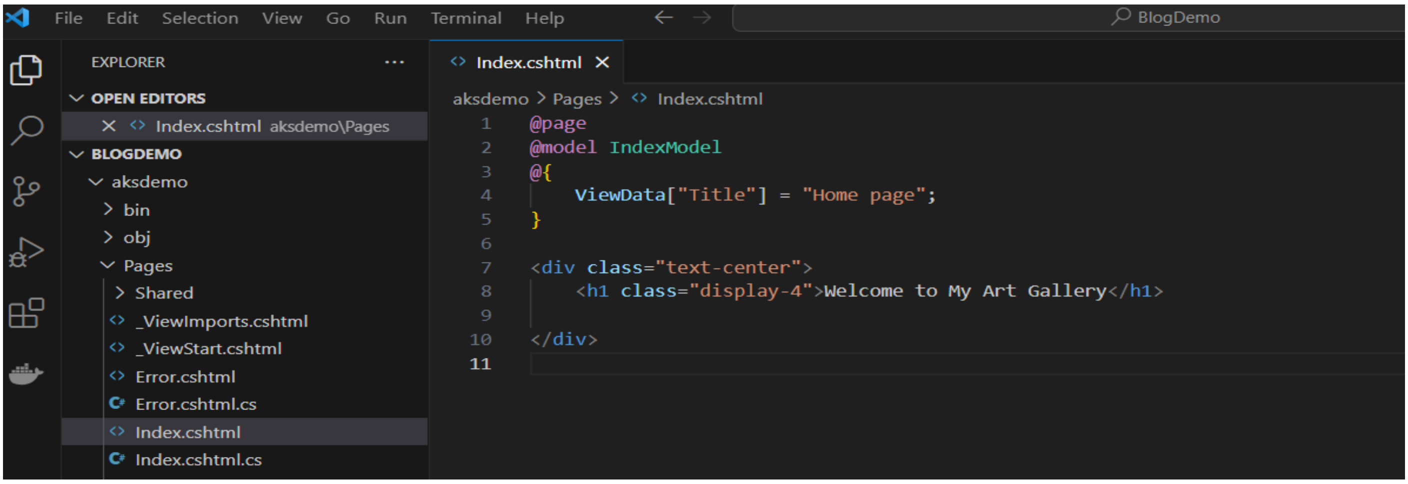
```
PS D:\Rahul\AzureDevOps\BlogDemo> dotnet new webapp --name aksdemo
The template "ASP.NET Core Web App" was created successfully.
PS D:\Rahul\AzureDevOps\BlogDemo> cd aksdemo
```

Build an image from Docker file using dotnet build command

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> dotnet build .
Build succeeded.
    0 Warning(s)
    0 Error(s)
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> dotnet run
PS D:\rahul\azuredevops\blogdemo\aksdemo> docker ps
```

Open Visual Studio code editor

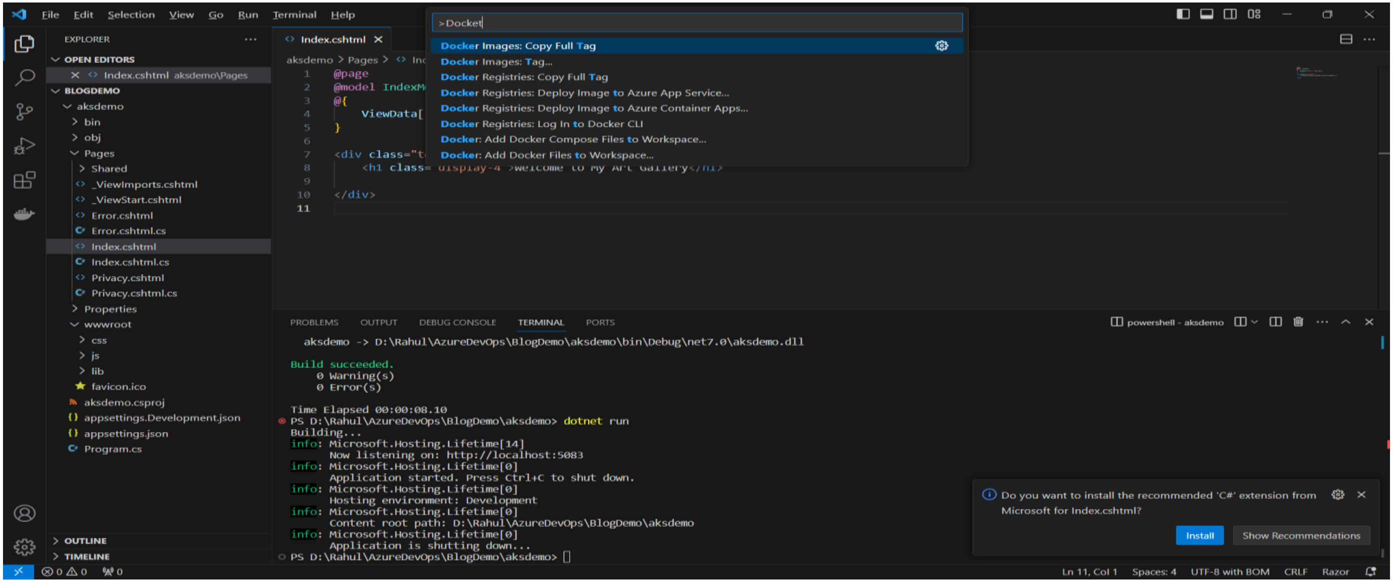
Open the folder Blogdemo and locate ***index.cshtml*** file and insert display text.



```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to My Art Gallery</h1>

</div>

```

In the Search Menu press Ctrl +P

Type Docker
Select Docker: Add Dockerfile to workspace
And select .net: asp.net core and the Port no

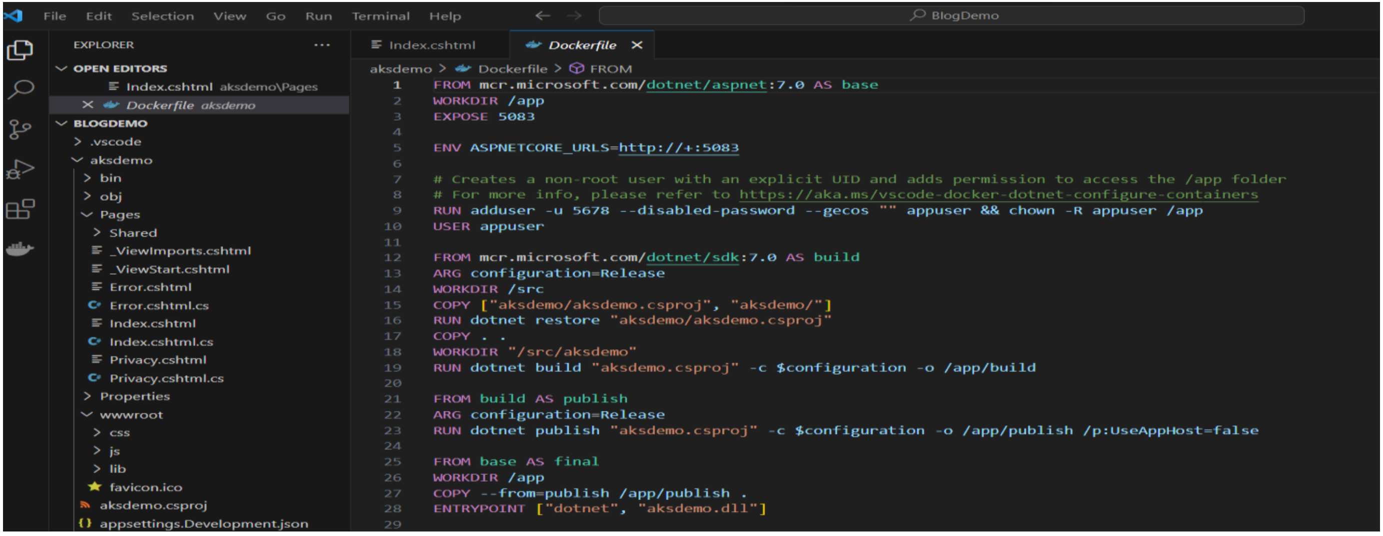


```
@page
@model IndexM
@{
    ViewData[
}

<div class="t
    <h1 class= display-4 >Welcome to My Art Gallery</h1>

</div>
```

```
  aksdemo -> D:\Rahul\AzureDevOps\BlogDemo\aksdemo\bin\Debug\net7.0\aksdemo.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:08.10
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5083
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\Rahul\AzureDevOps\BlogDemo\aksdemo
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> 
```

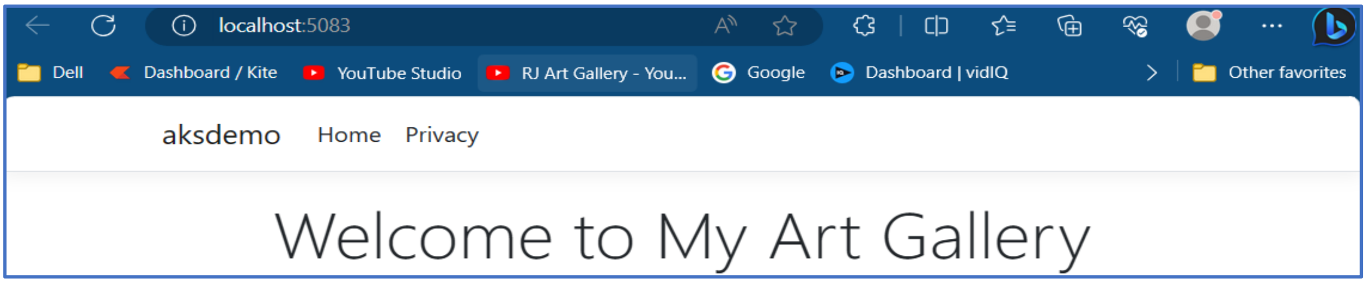
Do you want to install the recommended 'C#' extension from Microsoft for Index.cshtml?

Now you can see the Dockerfile is added to the workspace



```
FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
WORKDIR /app
EXPOSE 5083

ENV ASPNETCORE_URLS=http://+:5083

# Creates a non-root user with an explicit UID and adds permission to access the /app folder
# For more info, please refer to https://aka.ms/vscode-docker-dotnet-configure-containers
RUN adduser -u 5678 --disabled-password --gecos "" appuser && chown -R appuser /app
USER appuser

FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
ARG configuration=Release
WORKDIR /src
COPY ["aksdemo/aksdemo.csproj", "aksdemo/"]
RUN dotnet restore "aksdemo/aksdemo.csproj"
COPY . .
WORKDIR "/src/aksdemo"
RUN dotnet build "aksdemo.csproj" -c $configuration -o /app/build

FROM build AS publish
ARG configuration=Release
RUN dotnet publish "aksdemo.csproj" -c $configuration -o /app/publish /p:UseAppHost=false

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "aksdemo.dll"]
```

The Dockerfile is Created in aksdemo folder
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> ls Docker*
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo>docker images
Build Docker image in the folder where Docker file is present.
Move to the Parent folder
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> cd ..
PS D:\Rahul\AzureDevOps\BlogDemo> docker build -t aksdemo:v1.0 -f aksdemo/Dockerfile .
[+] Building 36.3s (19/19) FINISHED
S D:\Rahul\AzureDevOps\BlogDemo> cd aksdemo
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> dotnet run
Building...Now you can see asp.net application output in the Browser.



Welcome to My Art Gallery

Check the docker image status using the below command.

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> docker images aks*
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
aksdemo      v1.0      cfd5bf606738   21 minutes ago   222MB
```

Now login to Docker hub with your Credentials

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> docker login
Login Succeeded
```

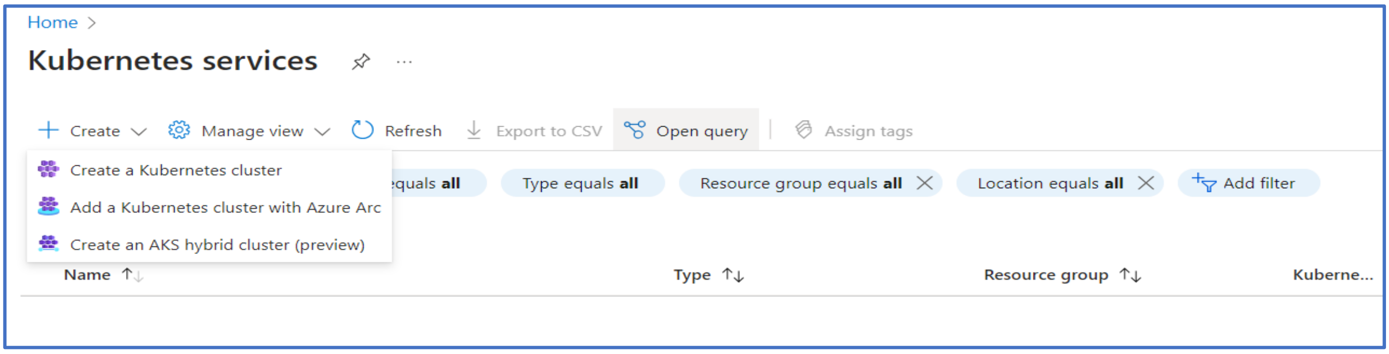
Create a repository in Docker Hub (my repository name is rahulazure98)

Now Build and push the image to Docker Hub

```
PS D:\Rahul\AzureDevOps\BlogDemo> docker build -t rahulazure98/aksdemo:v1.0 -f aksdemo/Dockerfile .
[+] Building 0.4s (19/19) FINISHED
PS D:\Rahul\AzureDevOps\BlogDemo> docker push rahulazure98/aksdemo:v1.0
The push refers to repository [docker.io/rahulazure98/aksdemo]
59c4c064f57c: Pushed
```

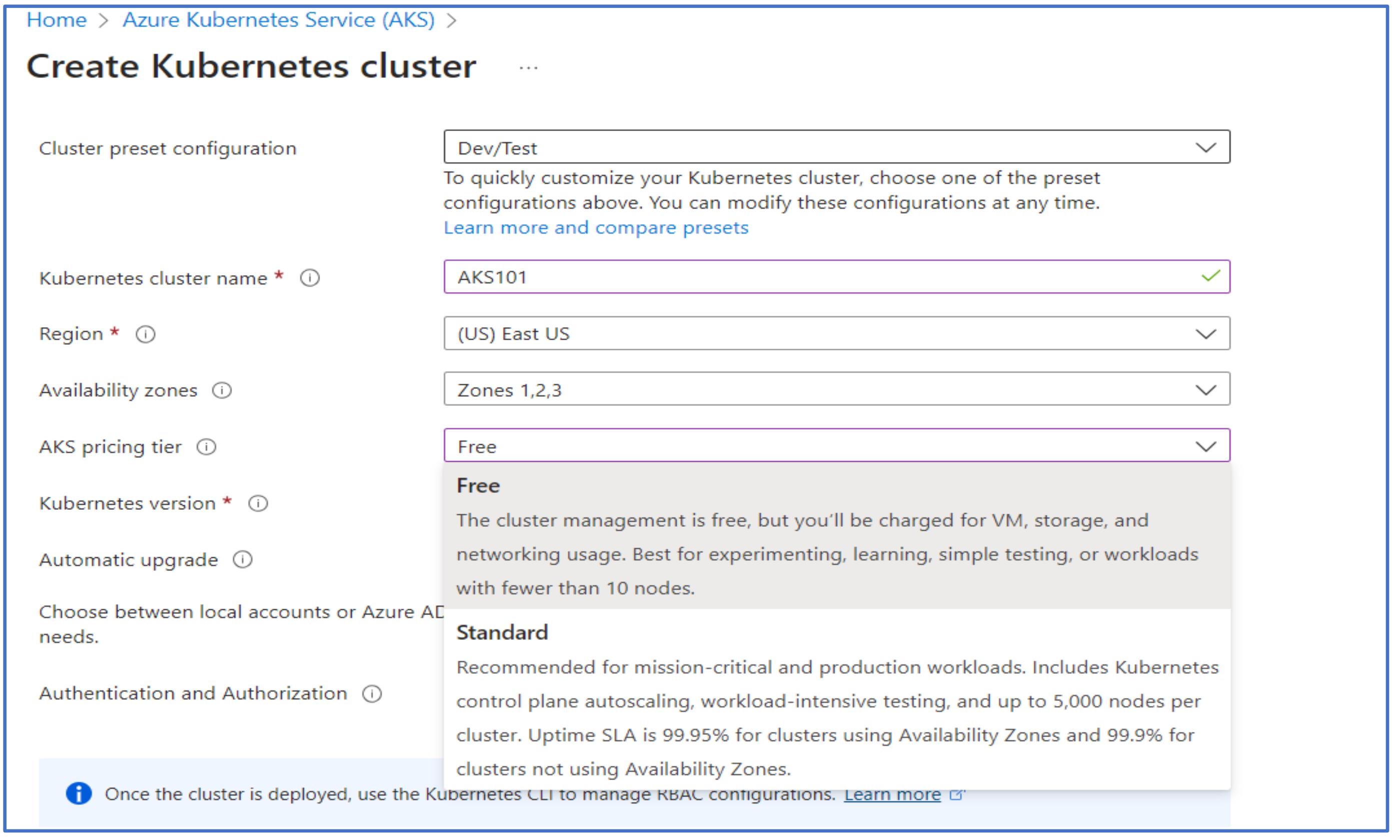
Create Azure Kubernetes Service from Azure Portal

In the azure portal, select Kubernetes Services



Select the Option Create a Kubernetes Cluster and Click Create Button

Provide the required details… Cluster name, Location, Free Tier



Create Kubernetes cluster

Cluster preset configuration: Dev/Test

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.
Learn more and compare presets

Kubernetes cluster name *: AKS101

Region *: (US) East US

Availability zones: Zones 1,2,3

AKS pricing tier: Free

Kubernetes version *

Automatic upgrade

Choose between local accounts or Azure AD needs.

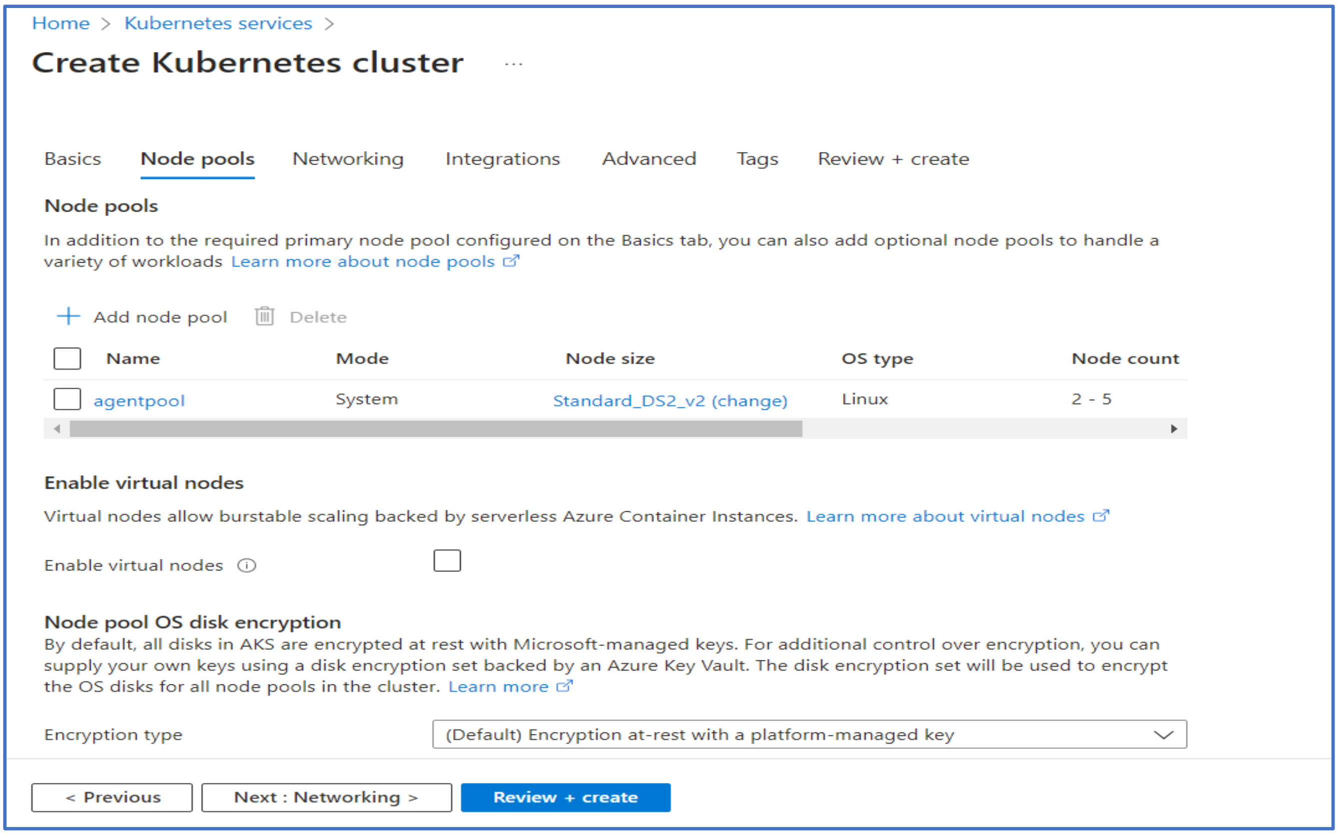
Authentication and Authorization

Free
The cluster management is free, but you'll be charged for VM, storage, and networking usage. Best for experimenting, learning, simple testing, or workloads with fewer than 10 nodes.

Standard
Recommended for mission-critical and production workloads. Includes Kubernetes control plane autoscaling, workload-intensive testing, and up to 5,000 nodes per cluster. Uptime SLA is 99.95% for clusters using Availability Zones and 99.9% for clusters not using Availability Zones.

Once the cluster is deployed, use the Kubernetes CLI to manage RBAC configurations. Learn more

Check the Node Pools tab



Home > Kubernetes services >

Create Kubernetes cluster ···

Basics | Node pools | Networking | Integrations | Advanced | Tags | Review + create

Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads Learn more about node pools

□ Name	Mode	Node size	OS type	Node count
□ agentpool	System	Standard_DS2_v2 (change)	Linux	2 - 5

Enable virtual nodes

Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. Learn more about virtual nodes

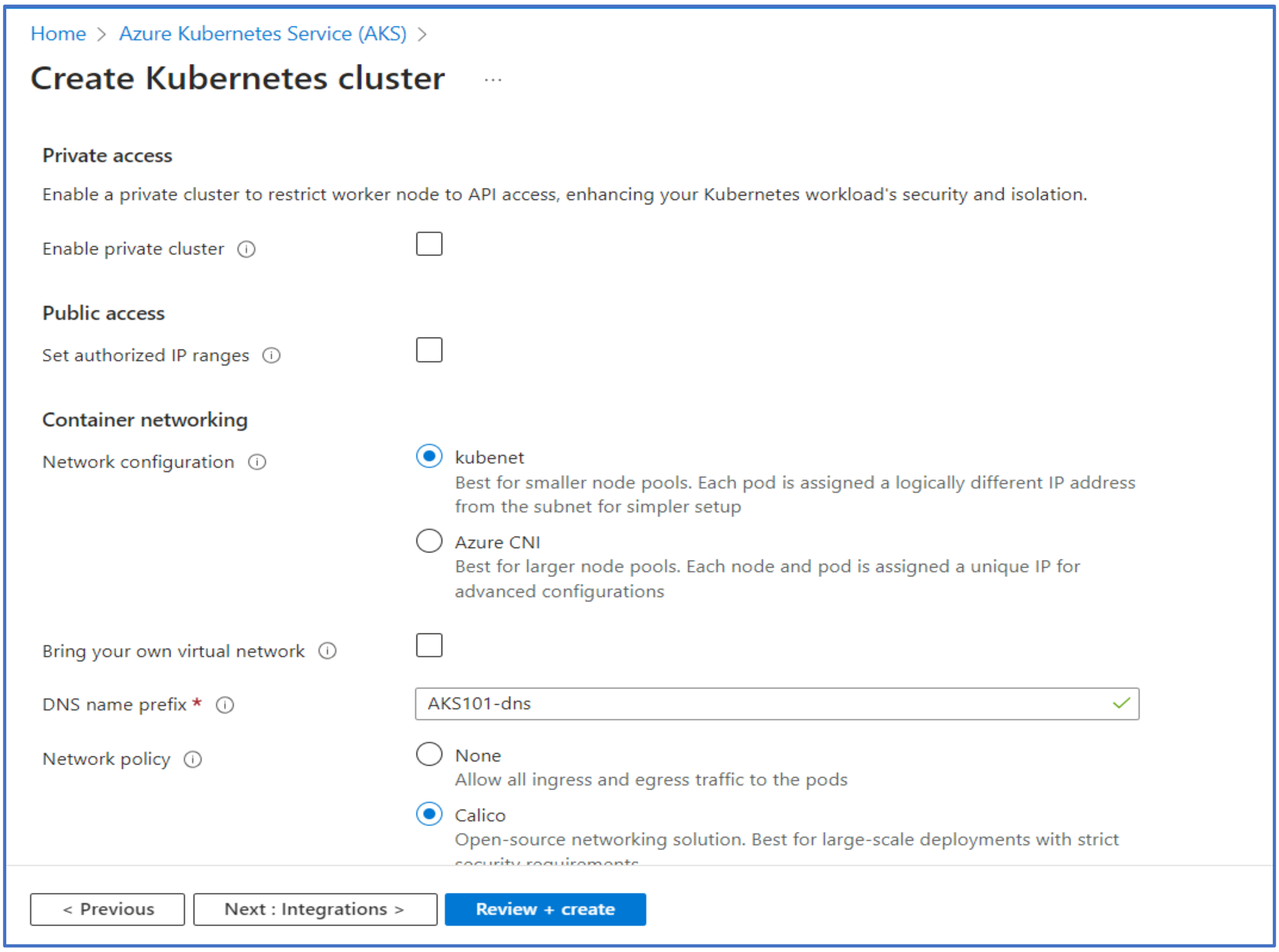
Enable virtual nodes ⓘ □

Node pool OS disk encryption

By default, all disks in AKS are encrypted at rest with Microsoft-managed keys. For additional control over encryption, you can supply your own keys using a disk encryption set backed by an Azure Key Vault. The disk encryption set will be used to encrypt the OS disks for all node pools in the cluster. Learn more

Encryption type: (Default) Encryption at-rest with a platform-managed key

Networking tab



Create Kubernetes cluster ···

Private access

Enable a private cluster to restrict worker node to API access, enhancing your Kubernetes workload's security and isolation.

Enable private cluster ⓘ □

Public access

Set authorized IP ranges ⓘ □

Container networking

Network configuration ⓘ

● kubenet
Best for smaller node pools. Each pod is assigned a logically different IP address from the subnet for simpler setup

○ Azure CNI
Best for larger node pools. Each node and pod is assigned a unique IP for advanced configurations

Bring your own virtual network ⓘ □

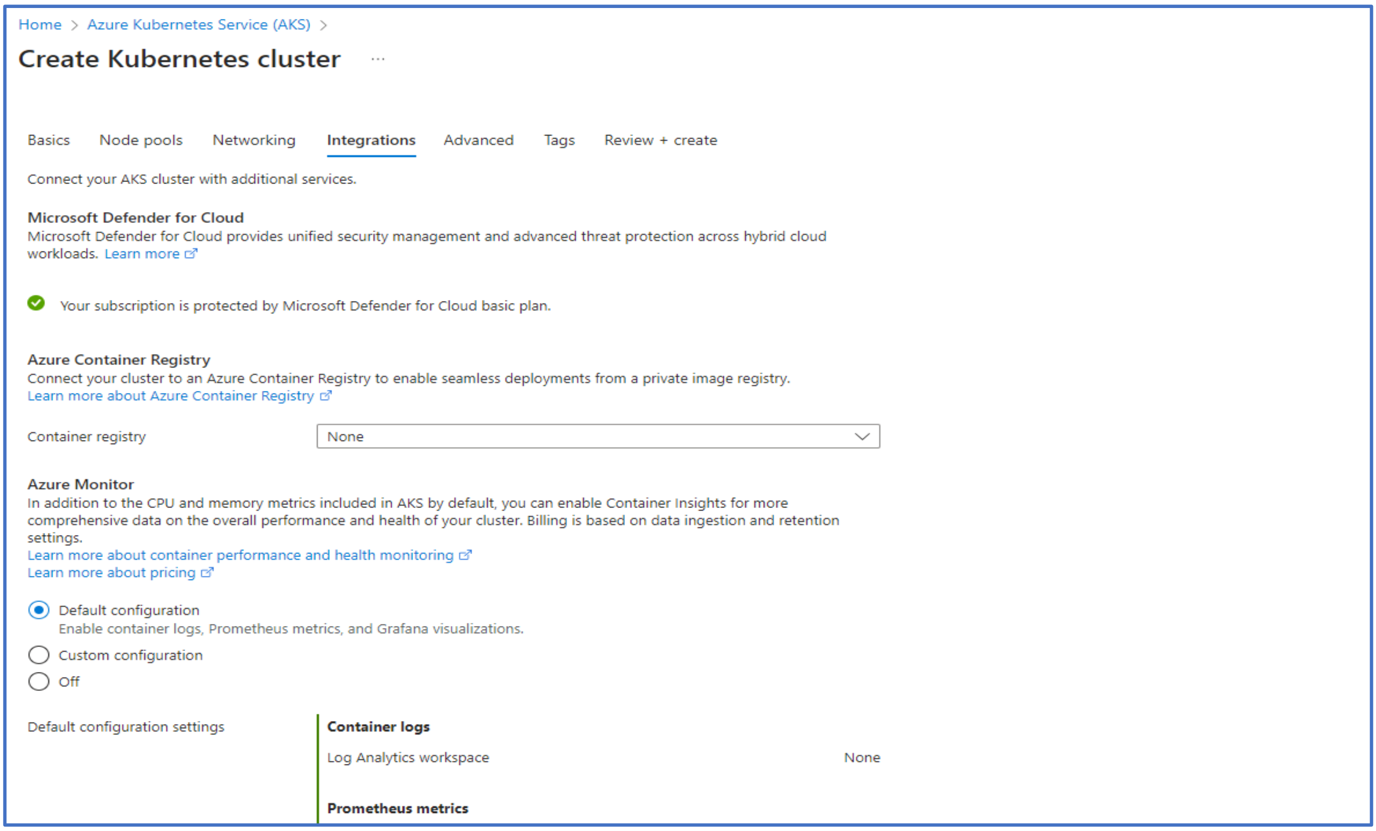
DNS name prefix * ⓘ AKS101-dns ✓

Network policy ⓘ

○ None
Allow all ingress and egress traffic to the pods

● Calico
Open-source networking solution. Best for large-scale deployments with strict security requirements

Integration tab



Create Kubernetes cluster ...

Connect your AKS cluster with additional services.

Microsoft Defender for Cloud
Microsoft Defender for Cloud provides unified security management and advanced threat protection across hybrid cloud workloads. Learn more

✓ Your subscription is protected by Microsoft Defender for Cloud basic plan.

Azure Container Registry
Connect your cluster to an Azure Container Registry to enable seamless deployments from a private image registry.
Learn more about Azure Container Registry

Container registry: None

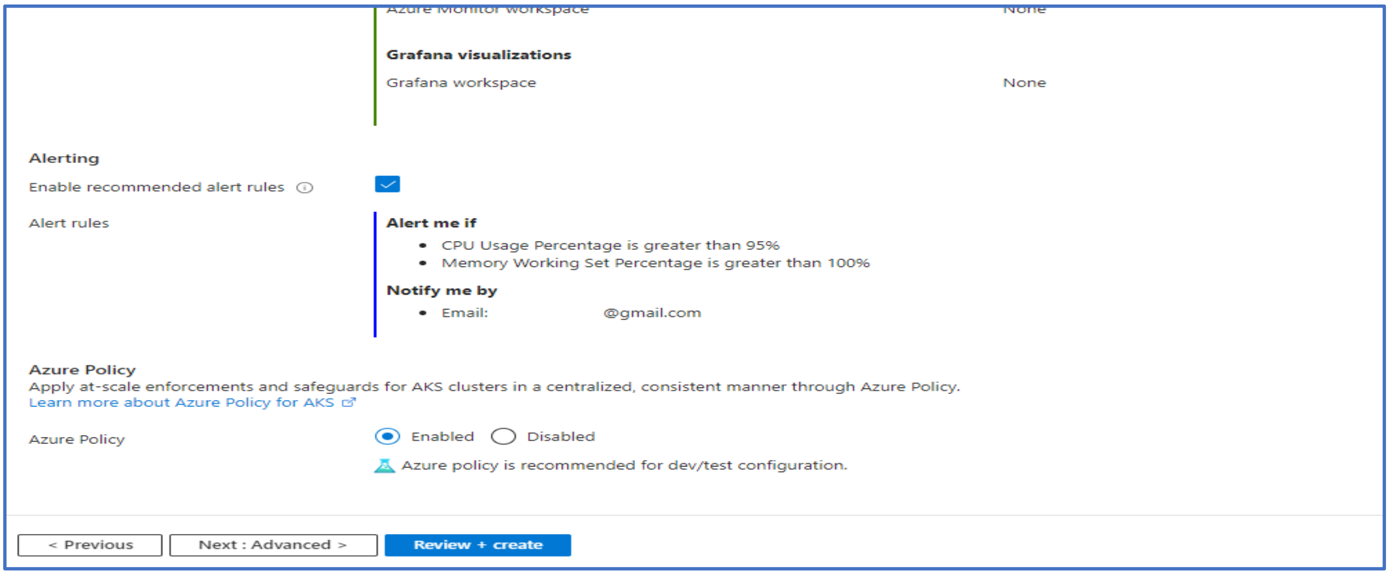
Azure Monitor
In addition to the CPU and memory metrics included in AKS by default, you can enable Container Insights for more comprehensive data on the overall performance and health of your cluster. Billing is based on data ingestion and retention settings.
Learn more about container performance and health monitoring
Learn more about pricing

● Default configuration
Enable container logs, Prometheus metrics, and Grafana visualizations.
○ Custom configuration
○ Off

Default configuration settings

Container logs
Log Analytics workspace: None

Prometheus metrics



Azure Monitor workspace: None

Grafana visualizations
Grafana workspace: None

Alerting
Enable recommended alert rules ⓘ ☑

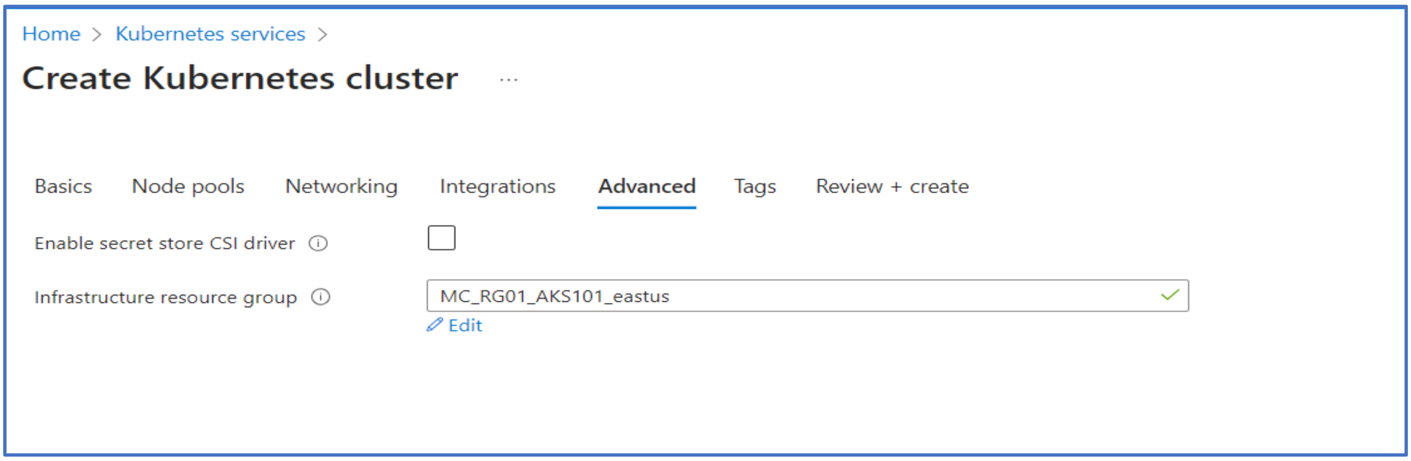
Alert rules

Alert me if
- CPU Usage Percentage is greater than 95%
- Memory Working Set Percentage is greater than 100%

Notify me by
- Email: @gmail.com

Azure Policy
Apply at-scale enforcements and safeguards for AKS clusters in a centralized, consistent manner through Azure Policy.
Learn more about Azure Policy for AKS

Azure Policy: ● Enabled ○ Disabled
Azure policy is recommended for dev/test configuration.

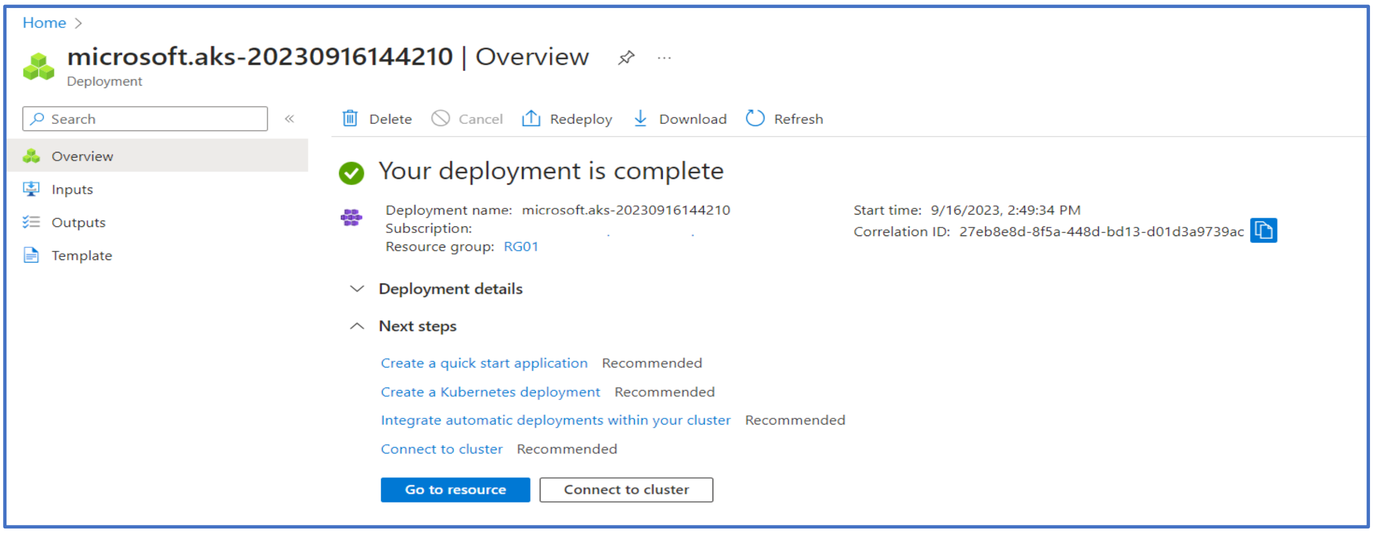


Create Kubernetes cluster ...

Enable secret store CSI driver ⓘ □

Infrastructure resource group ⓘ: MC_RG01_AKS101_eastus ✓
Edit

Select Review and Create.



Home >

microsoft.aks-20230916144210 | Overview

Deployment

Your deployment is complete

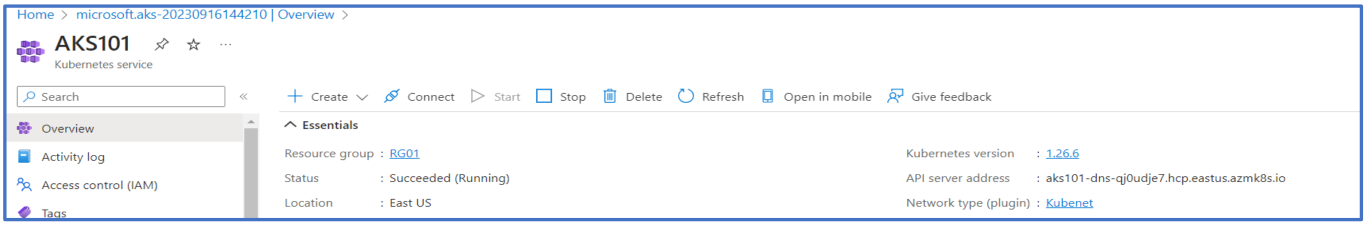
Deployment name: microsoft.aks-20230916144210
Subscription:
Resource group: RG01

Start time: 9/16/2023, 2:49:34 PM
Correlation ID: 27eb8e8d-8f5a-448d-bd13-d01d3a9739ac

Deployment details

Next steps

Create a quick start application Recommended
Create a Kubernetes deployment Recommended
Integrate automatic deployments within your cluster Recommended
Connect to cluster Recommended



Home > microsoft.aks-20230916144210 | Overview >

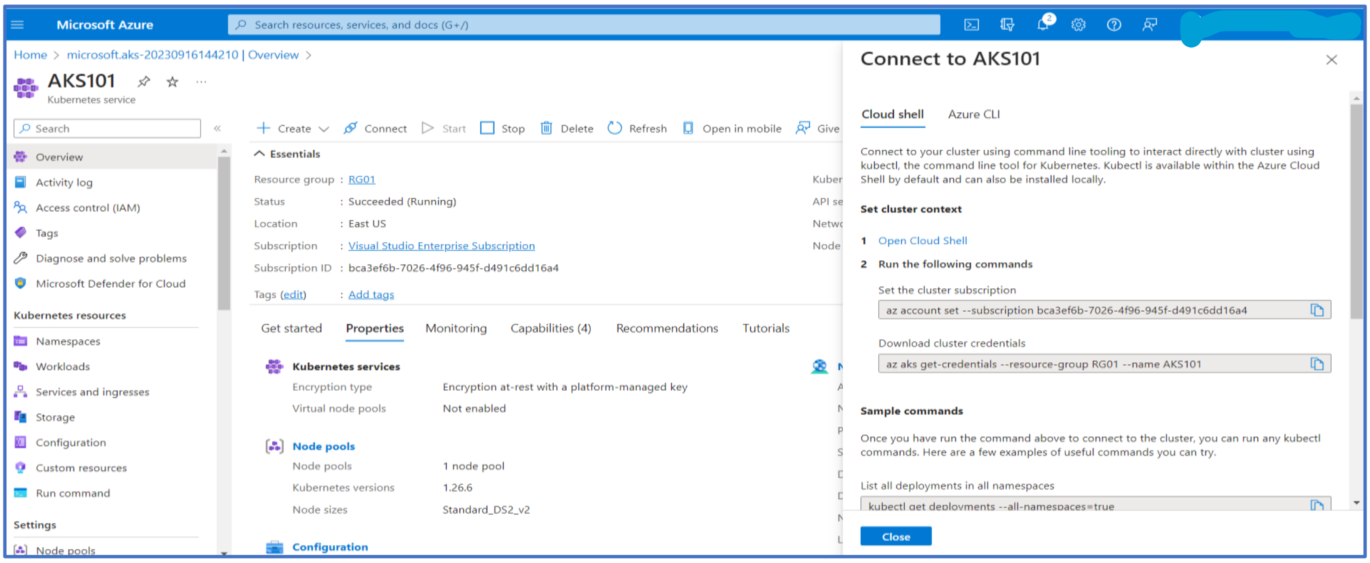
AKS101

Kubernetes service

Essentials

Resource group : RG01
Status : Succeeded (Running)
Location : East US

Kubernetes version : 1.26.6
API server address : aks101-dns-qj0udje7.hcp.eastus.azmk8s.io
Network type (plugin) : Kubenet



[redacted]

Home > microsoft.aks-20230916144210 | Overview >

AKS101

Kubernetes service

Essentials

Resource group : RG01
Status : Succeeded (Running)
Location : East US
Subscription : Visual Studio Enterprise Subscription
Subscription ID : bca3ef6b-7026-4f96-945f-d491c6dd16a4
Tags (edit) : Add tags

Kubernetes services
Encryption type Encryption at-rest with a platform-managed key
Virtual node pools Not enabled

Node pools
Node pools 1 node pool
Kubernetes versions 1.26.6
Node sizes Standard_DS2_v2

Configuration

Connect to AKS101

Connect to your cluster using command line tooling to interact directly with cluster using kubectl, the command line tool for Kubernetes. Kubectl is available within the Azure Cloud Shell by default and can also be installed locally.

Set cluster context

1 Open Cloud Shell

2 Run the following commands

Set the cluster subscription

```
az account set --subscription bca3ef6b-7026-4f96-945f-d491c6dd16a4
```

Download cluster credentials

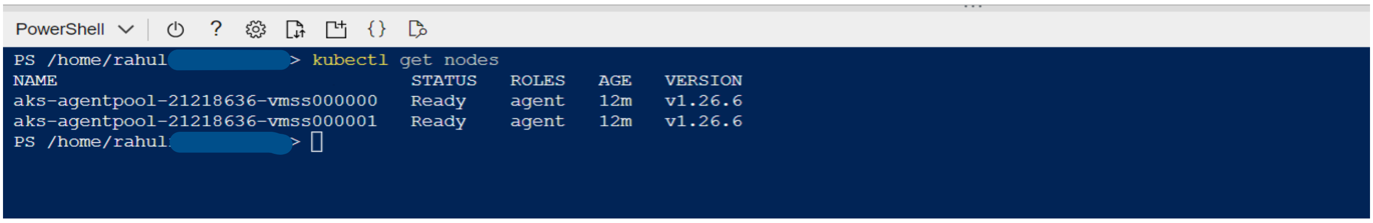
```
az aks get-credentials --resource-group RG01 --name AKS101
```

Sample commands

Once you have run the command above to connect to the cluster, you can run any kubectl commands. Here are a few examples of useful commands you can try.

List all deployments in all namespaces

```
kubectl get deployments --all-namespaces=true
```



```
PS /home/rahul[redacted]> kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-21218636-vmss000000   Ready    agent   12m   v1.26.6
aks-agentpool-21218636-vmss000001   Ready    agent   12m   v1.26.6
PS /home/rahul[redacted]> 
```

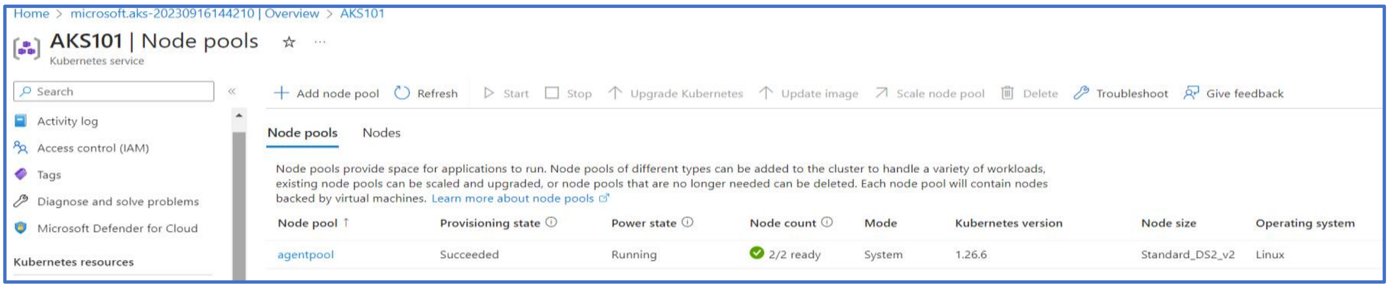
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> az login
A web browser has been opened and logged in
Copy & Paste the Cluster Subscription text
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> az account set --subscription bca3ef6b-7026-4f96-945f-d491c6dd16a4
Copy & Paste the Cluster Credentials text
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> az aks get-credentials --resource-group RG01 --name AKS101

PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> kubectl create deploy aksdemo101 --image=rahulazure98/aksdemo:v1
deployment.apps/aksdemo101 created

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> kubectl get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
aksdemo101   0/1     1            0           17s
```

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> kubectl get pods
NAME                       READY   STATUS          RESTARTS   AGE
aksdemo-5fb5fb4d99-2cqpb     1/1     Running           0         55s
```

```
PS D:\Rahul\AzureDevOps\BlogDemo\aksdemo> kubectl get svc -w
NAME       TYPE          CLUSTER-IP   EXTERNAL-IP    PORT(S)       AGE
aksdemo     LoadBalancer   10.0.160.8   104.45.169.73   80:32161/TCP   21s
```



Home > microsoft.aks-20230916144210 | Overview > AKS101

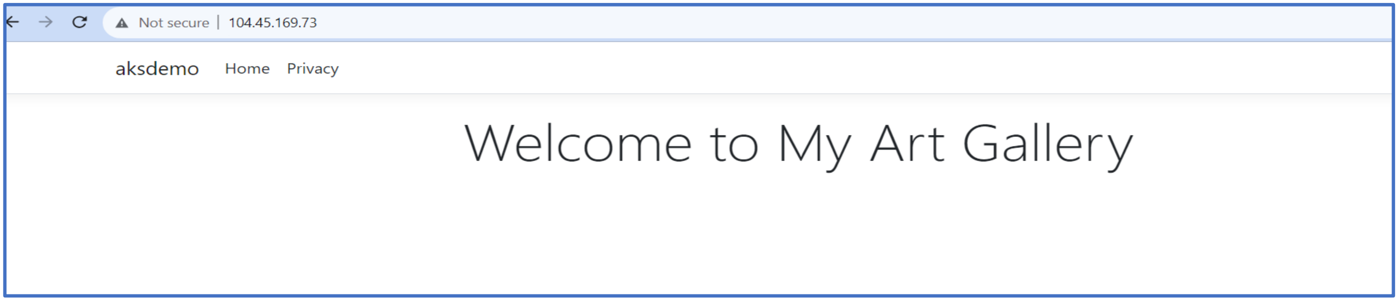
AKS101 | Node pools ☆ ···

Kubernetes service

Node pools provide space for applications to run. Node pools of different types can be added to the cluster to handle a variety of workloads, existing node pools can be scaled and upgraded, or node pools that are no longer needed can be deleted. Each node pool will contain nodes backed by virtual machines. Learn more about node pools

Node pool ↑	Provisioning state	Power state	Node count	Mode	Kubernetes version	Node size	Operating system
agentpool	Succeeded	Running	✓ 2/2 ready	System	1.26.6	Standard_DS2_v2	Linux

Copy and paste the External IP in the Browse

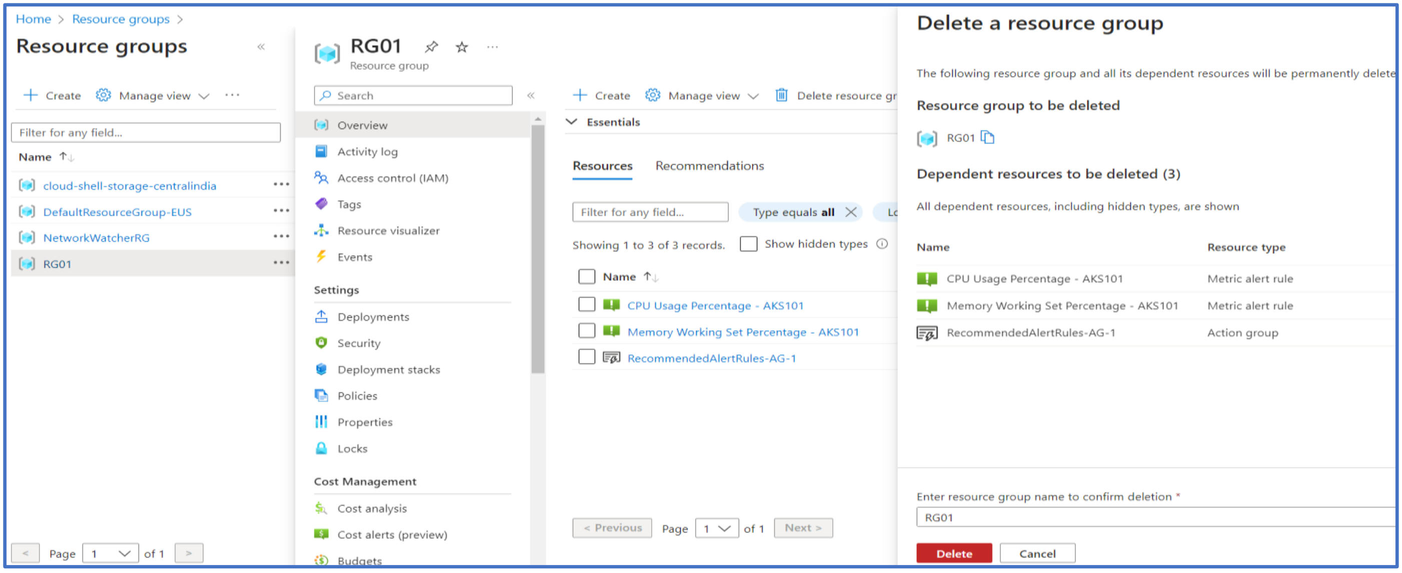


Not secure | 104.45.169.73

Welcome to My Art Gallery

Cleanup Resources

Delete Resource Group.



Home > Resource groups >

Resource groups

cloud-shell-storage-centralindia

DefaultResourceGroup-EUS

NetworkWatcherRG

RG01

RG01

Resource group

Showing 1 to 3 of 3 records. □ Show hidden types

	Name ↑↓
□	CPU Usage Percentage - AKS101
□	Memory Working Set Percentage - AKS101
□	RecommendedAlertRules-AG-1

Delete a resource group

The following resource group and all its dependent resources will be permanently delete

Resource group to be deleted

RG01

Dependent resources to be deleted (3)

All dependent resources, including hidden types, are shown

Name	Resource type
CPU Usage Percentage - AKS101	Metric alert rule
Memory Working Set Percentage - AKS101	Metric alert rule
RecommendedAlertRules-AG-1	Action group

Enter resource group name to confirm deletion *

RG01