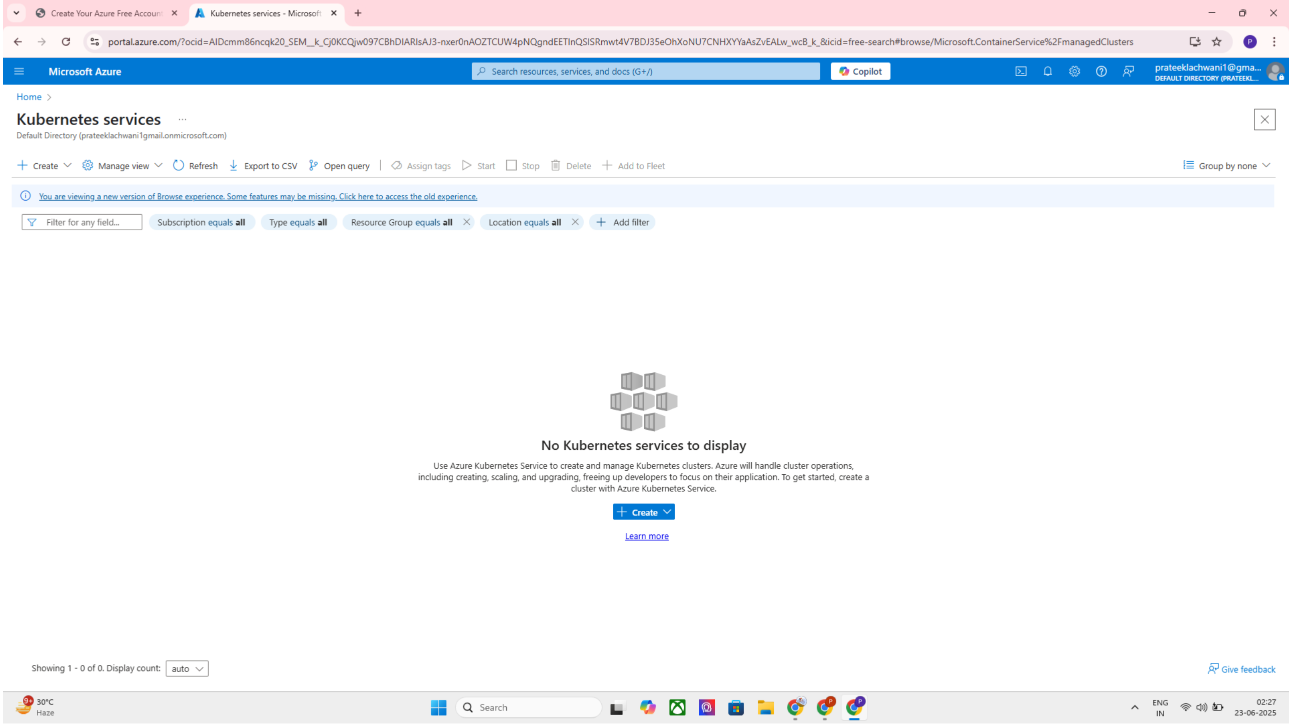
Ques 3 : **Deploy an AKS cluster using the portal. Access the dashboard and create roles for multiple users**

SOLN:

GO TO AZURE KUBERNETES CONTAINER SERVICES AND CREATE ONE :


Create Your Azure Free Account
Kubernetes services - Microsoft
portal.azure.com/?ocid=AIDcmm86ncqk20_SEM__k_Cj0KCQjw097CBhDIARIsAJ3-nxer0nAOZTCUW4pNQgndEETlnQSISRmwt4V7BDJ35eOhXoNU7CNHXYYaAsZvEALw_wcB_k_&icid=free-search#browse/Microsoft.ContainerService%2FmanagedClusters
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateeklachwani1@gma...
DEFAULT DIRECTORY (PRATEEKL...
Home
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Start
Stop
Delete
Add to Fleet
Group by none
Filter for any field...
Subscription equals all
Type equals all
Resource Group equals all
Location equals all
Add filter
Create
Learn more
Give feedback
30°C
Haze
Search
ENG
IN
02:27
23-06-2025

Kubernetes services

Default Directory (prateeklachwani1gmail.onmicrosoft.com)

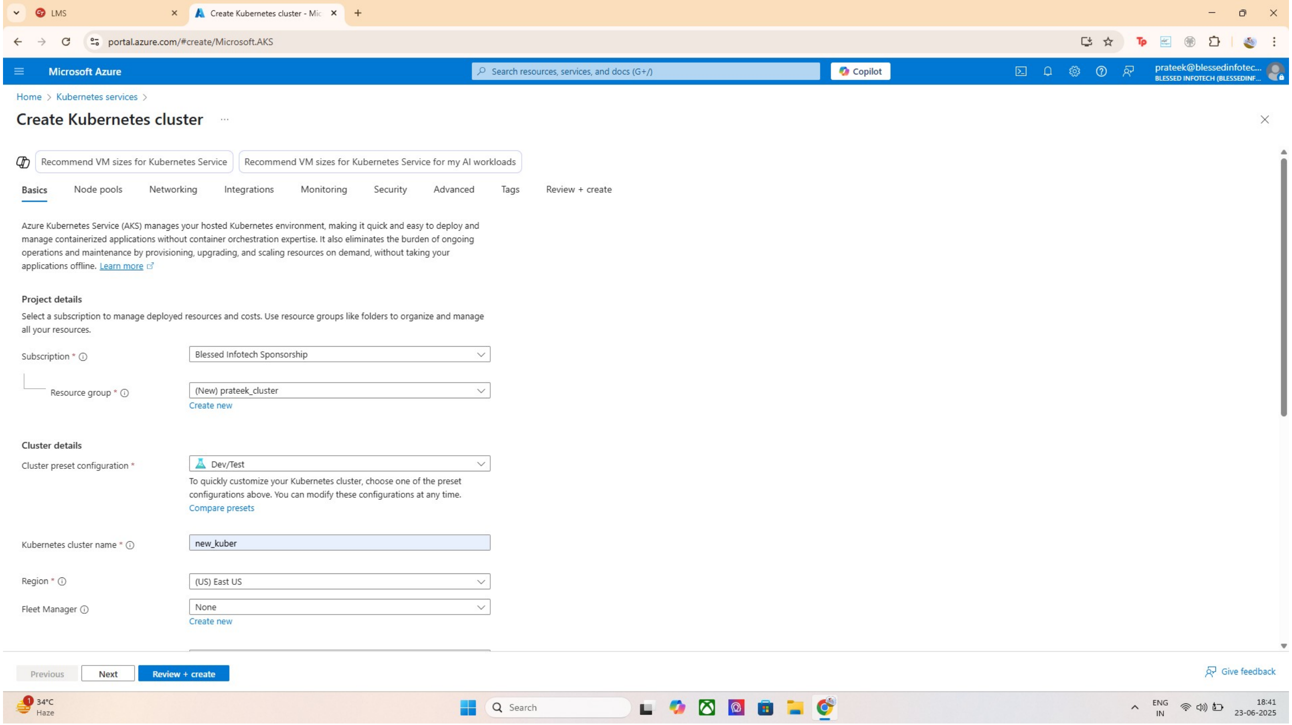
You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

No Kubernetes services to display

Use Azure Kubernetes Service to create and manage Kubernetes clusters. Azure will handle cluster operations, including creating, scaling, and upgrading, freeing up developers to focus on their application. To get started, create a cluster with Azure Kubernetes Service.

Showing 1 - 0 of 0. Display count: auto

MAKE SURE TO VERIFY THE CONFIGURATIONS :


LMS
Create Kubernetes cluster - Mic
portal.azure.com/#create/Microsoft.AKS
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateek@blessedinfotec...
BLESSED INFOTECH (BLESSEDINF...
Home
Kubernetes services
Recommend VM sizes for Kubernetes Service
Recommend VM sizes for Kubernetes Service for my AI workloads
Basics
Node pools
Networking
Integrations
Monitoring
Security
Advanced
Tags
Review + create
Previous
Next
Review + create
Give feedback
34°C
Haze
Search
ENG
IN
18:41
23-06-2025

Create Kubernetes cluster

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. Learn more

Project details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Blessed Infotech Sponsorship

Resource group * : (New) prateek_cluster

Create new

Cluster details

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.

Compare presets

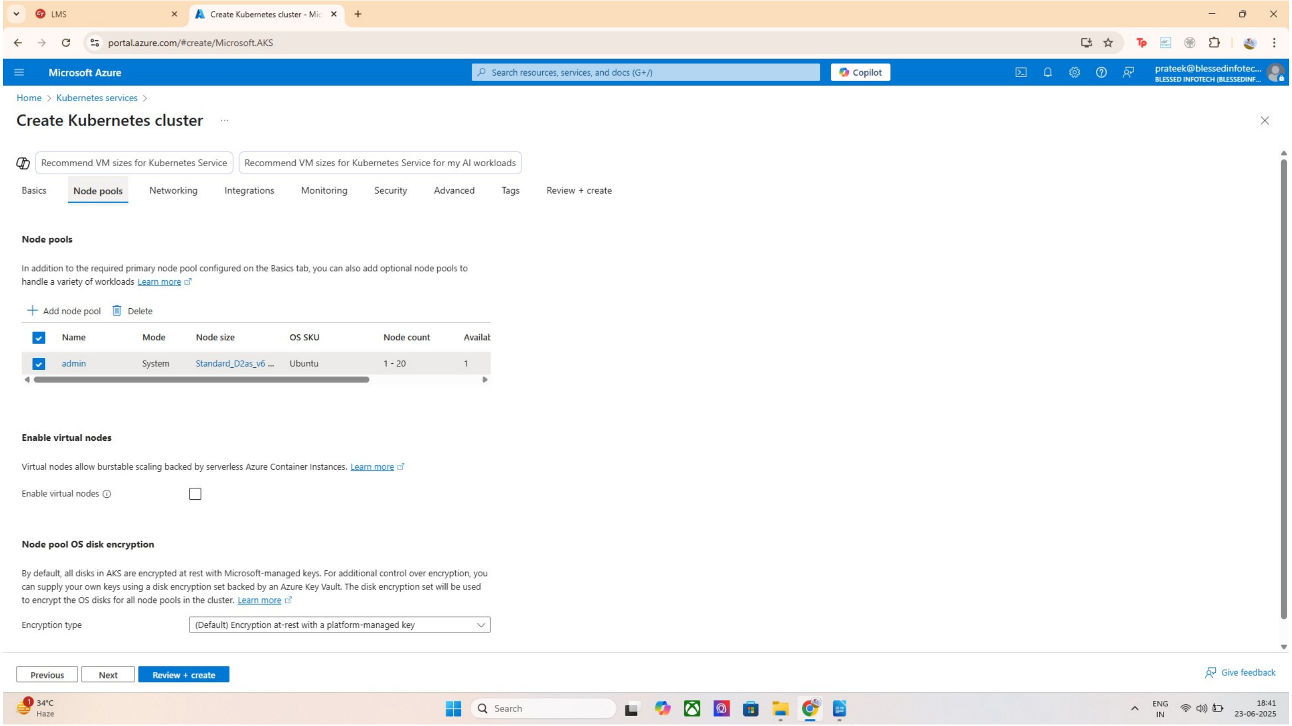
Kubernetes cluster name * : new_kuber

Region * : (US) East US

Fleet Manager : None

Create new

SET THE NODE PODS AND THE CONTAINER VM CONFIGURATION AS PER THE ZONE :



LMS
Create Kubernetes cluster - Mic
portal.azure.com/#create/Microsoft.AKS
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateek@blessedinfotec...
BLESSED INFOTECH (BLESSEDINF...
Home > Kubernetes services >
Recommend VM sizes for Kubernetes Service
Recommend VM sizes for Kubernetes Service for my AI workloads
Basics
Node pools
Networking
Integrations
Monitoring
Security
Advanced
Tags
Review + create
Add node pool
Delete
Previous
Next
Review + create
Give feedback
34°C
Haze
Search
ENG
IN
18:41
23-06-2025

Create Kubernetes cluster

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

☑	Name	Mode	Node size	OS SKU	Node count	Availat
☑	admin	System	Standard_D2as_v6 ...	Ubuntu	1 - 20	1

Enable virtual nodes

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

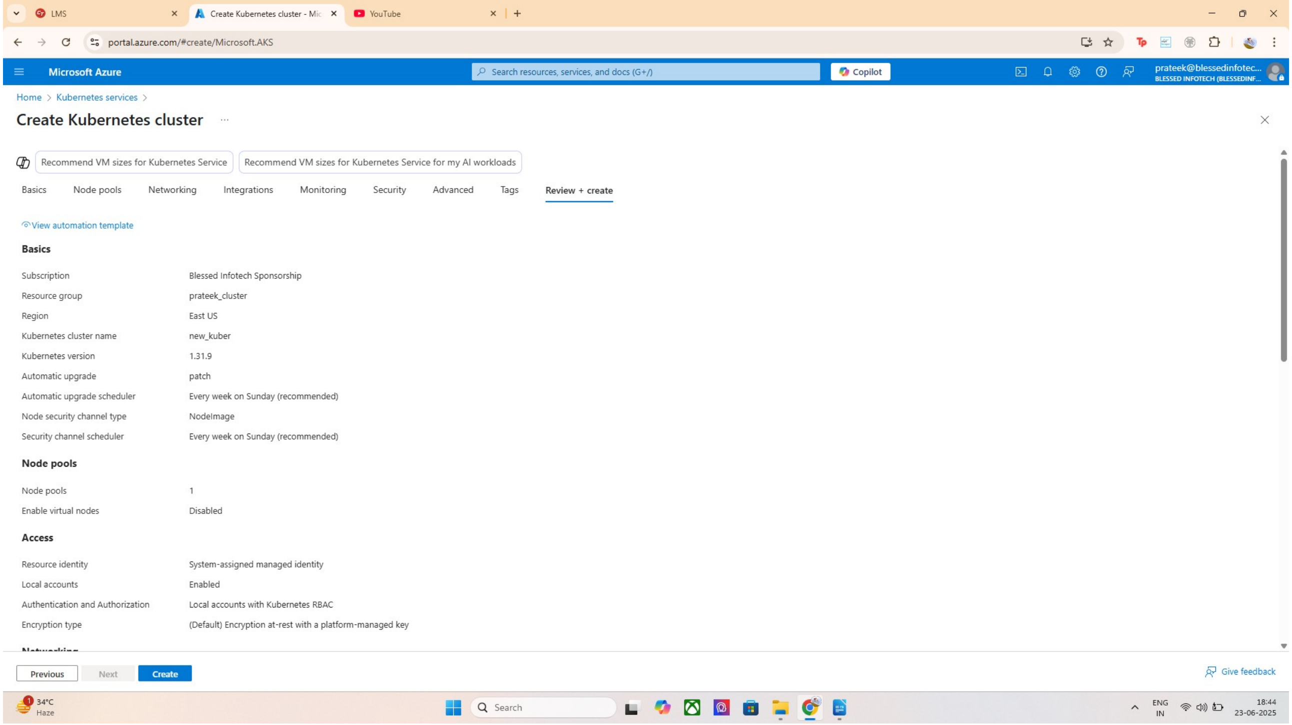
Enable virtual nodes ⓘ □

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

REVEIW AND CREATE A KUBERNETES CLUSTER



LMS
Create Kubernetes cluster - Mic
YouTube
portal.azure.com/#create/Microsoft.AKS
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateek@blessedinfotec...
BLESSED INFOTECH (BLESSEDINF...
Home > Kubernetes services >
Recommend VM sizes for Kubernetes Service
Recommend VM sizes for Kubernetes Service for my AI workloads
Basics
Node pools
Networking
Integrations
Monitoring
Security
Advanced
Tags
Review + create
View automation template
Previous
Next
Create
Give feedback
34°C
Haze
Search
ENG
IN
18:44
23-06-2025

Create Kubernetes cluster

Basics

Subscription	Blessed Infotech Sponsorship
Resource group	prateek_cluster
Region	East US
Kubernetes cluster name	new_kuber
Kubernetes version	1.31.9
Automatic upgrade	patch
Automatic upgrade scheduler	Every week on Sunday (recommended)
Node security channel type	NodeImage
Security channel scheduler	Every week on Sunday (recommended)

Node pools

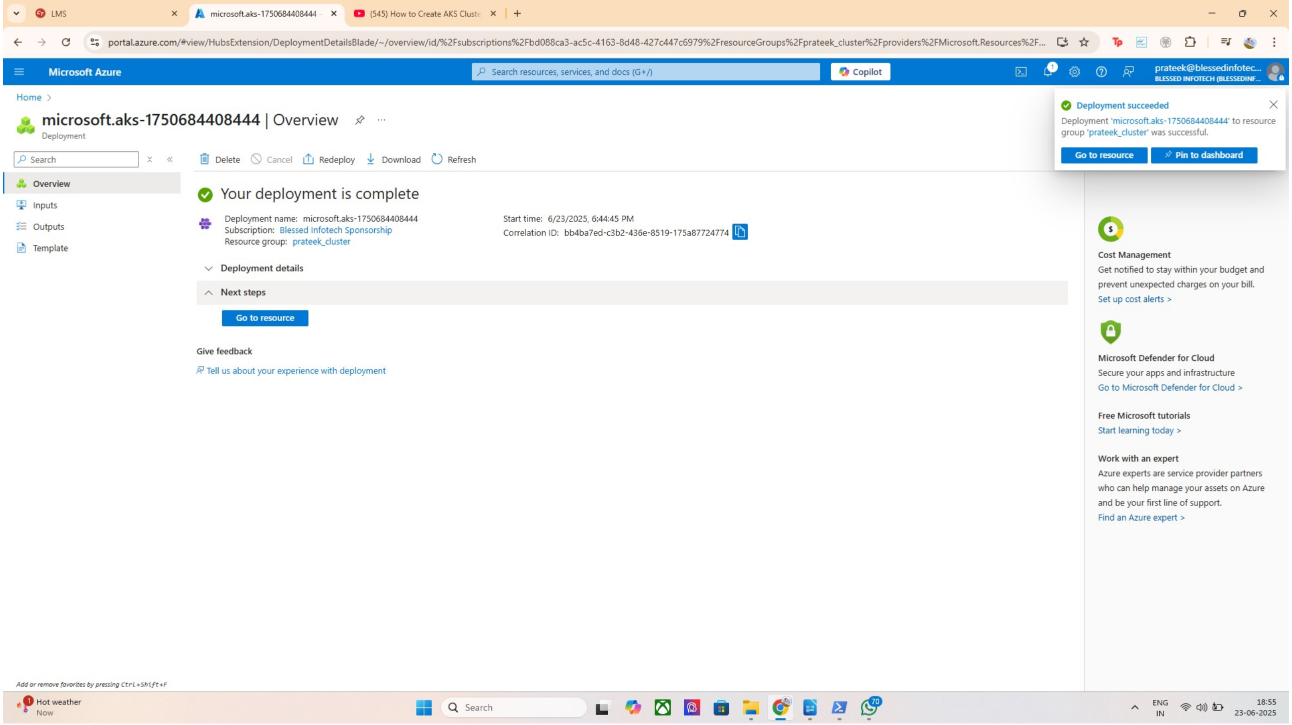
Node pools	1
Enable virtual nodes	Disabled

Access

Resource identity	System-assigned managed identity
Local accounts	Enabled
Authentication and Authorization	Local accounts with Kubernetes RBAC
Encryption type	(Default) Encryption at-rest with a platform-managed key

Networking

VERIFY THE DEPLOYMENT:


LMS
microsoft.aks-1750684408444
(545) How to Create AKS Cluste
portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Fbd088ca3-ac5c-4163-8d48-427c447c6979%2FresourceGroups%2Fprateek_cluster%2Fproviders%2FMicrosoft.Resources%2F...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateek@blessedinfotec...
BLESSED INFOTECH (BLESSEDINF...
Home
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Go to resource
Pin to dashboard
Add or remove favorites by pressing Ctrl+Shift+F
Hot weather
Now
Search
ENG
IN
18:55
23-06-2025

microsoft.aks-1750684408444 | Overview

Deployment

Your deployment is complete

Deployment name: microsoft.aks-1750684408444
Subscription: Blessed Infotech Sponsorship
Resource group: prateek_cluster

Start time: 6/23/2025, 6:44:45 PM
Correlation ID: bb4ba7ed-c3b2-436e-8519-175a87724774

Deployment details

Next steps

Give feedback

Tell us about your experience with deployment

Deployment succeeded

Deployment 'microsoft.aks-1750684408444' to resource group 'prateek_cluster' was successful.

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 >

Free Microsoft tutorials

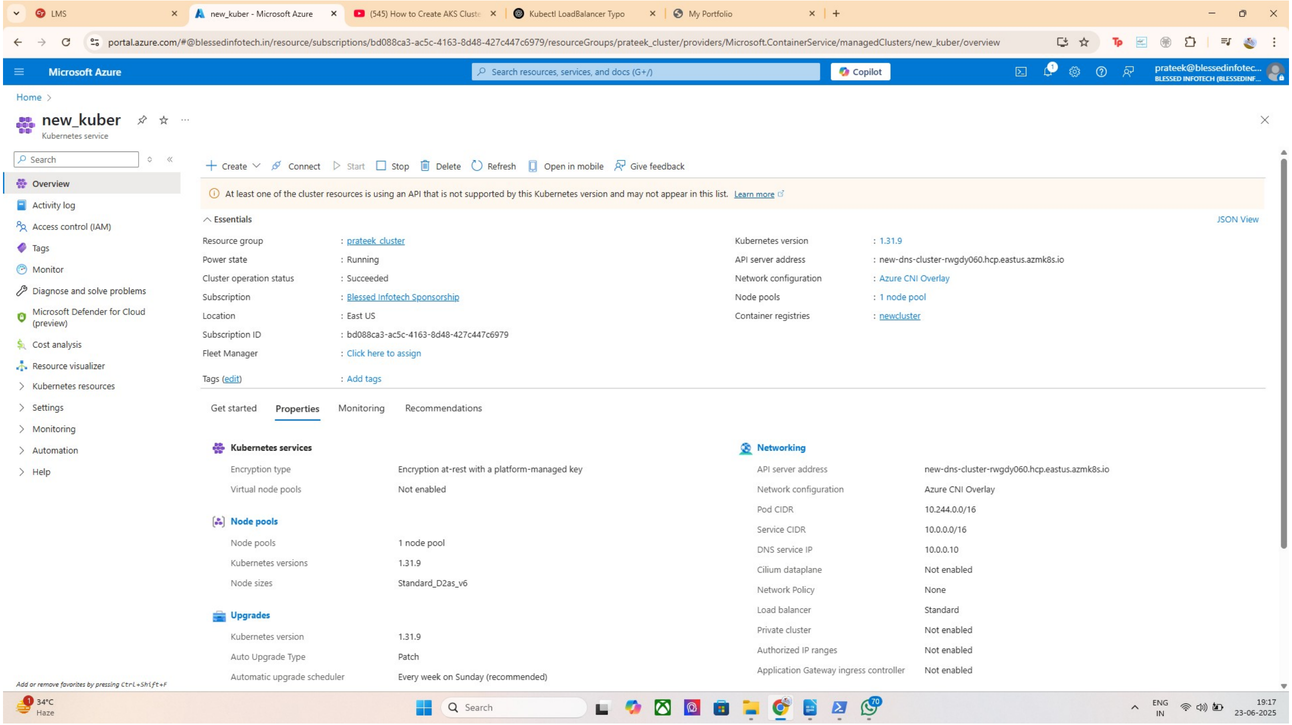
Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

CREATED AND ACCESSED THE NEW_CLUSTER


LMS
new_kuber - Microsoft Azure
(545) How to Create AKS Cluste
Kubectl LoadBalancer Typo
My Portfolio
portal.azure.com/#@blessedinfotech.in/resource/subscriptions/bd088ca3-ac5c-4163-8d48-427c447c6979/resourceGroups/prateek_cluster/providers/Microsoft.ContainerService/managedClusters/new_kuber/overview
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
prateek@blessedinfotec...
BLESSED INFOTECH (BLESSEDINF...
Home
Search
Overview
Activity log
Access control (IAM)
Tags
Monitor
Diagnose and solve problems
Microsoft Defender for Cloud (preview)
Cost analysis
Resource visualizer
Kubernetes resources
Settings
Monitoring
Automation
Help
Create
Connect
Start
Stop
Delete
Refresh
Open in mobile
Give feedback
JSON View
Get started
Properties
Monitoring
Recommendations
Add or remove favorites by pressing Ctrl+Shift+F
34°C
Haze
Search
ENG
IN
19:17
23-06-2025

new_kuber

Kubernetes service

At least one of the cluster resources is using an API that is not supported by this Kubernetes version and may not appear in this list. Learn more

Essentials

Resource group	: prateek_cluster
Power state	: Running
Cluster operation status	: Succeeded
Subscription	: Blessed Infotech Sponsorship
Location	: East US
Subscription ID	: bd088ca3-ac5c-4163-8d48-427c447c6979
Fleet Manager	: Click here to assign
Tags (edit)	: Add tags

Kubernetes version	: 1.31.9
API server address	: new-dns-cluster-rwgdy060.hcp.eastus.azmk8s.io
Network configuration	: Azure CNI Overlay
Node pools	: 1 node pool
Container registries	: newcluster

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	1.31.9
Node sizes	Standard_D2as_v6

Upgrades

Kubernetes version	1.31.9
Auto Upgrade Type	Patch
Automatic upgrade scheduler	Every week on Sunday (recommended)

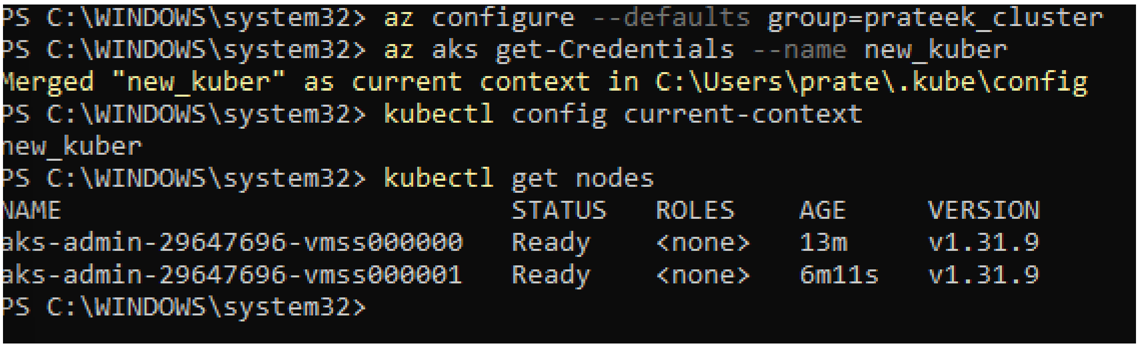
Networking

API server address	new-dns-cluster-rwgdy060.hcp.eastus.azmk8s.io
Network configuration	Azure CNI Overlay
Pod CIDR	10.244.0.0/16
Service CIDR	10.0.0.0/16
DNS service IP	10.0.0.10
Cilium dataplane	Not enabled
Network Policy	None
Load balancer	Standard
Private cluster	Not enabled
Authorized IP ranges	Not enabled
Application Gateway ingress controller	Not enabled

COMMAND USED : az configure --defaults group=prateek_cluster
az aks get-credentials --name new_kuber
kubectl config current-context
kubectl get nodes

`az configure --defaults group=prateek_cluster`
➤ Sets the default Azure resource group to `prateek_cluster` for future `az` commands.

- `az aks get-credentials --name new_kuber`
  ➤ Downloads the kubeconfig for the `new_kuber` AKS cluster and merges it into your local `~/.kube/config`.
- `kubectl config current-context`
  ➤ Verifies that your `kubectl` is now set to use the `new_kuber` cluster.
- `kubectl get nodes`
  ➤ Lists all nodes in the cluster, confirming that the cluster is running and ready.

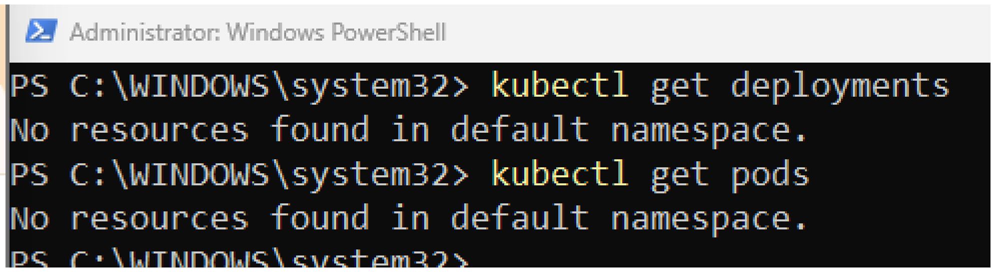


```
PS C:\WINDOWS\system32> az configure --defaults group=prateek_cluster
PS C:\WINDOWS\system32> az aks get-Credentials --name new_kuber
Merged "new_kuber" as current context in C:\Users\prate\.kube\config
PS C:\WINDOWS\system32> kubectl config current-context
new_kuber
PS C:\WINDOWS\system32> kubectl get nodes
NAME                                STATUS   ROLES    AGE     VERSION
aks-admin-29647696-vmss000000   Ready    <none>   13m     v1.31.9
aks-admin-29647696-vmss000001   Ready    <none>   6m11s   v1.31.9
PS C:\WINDOWS\system32>
```

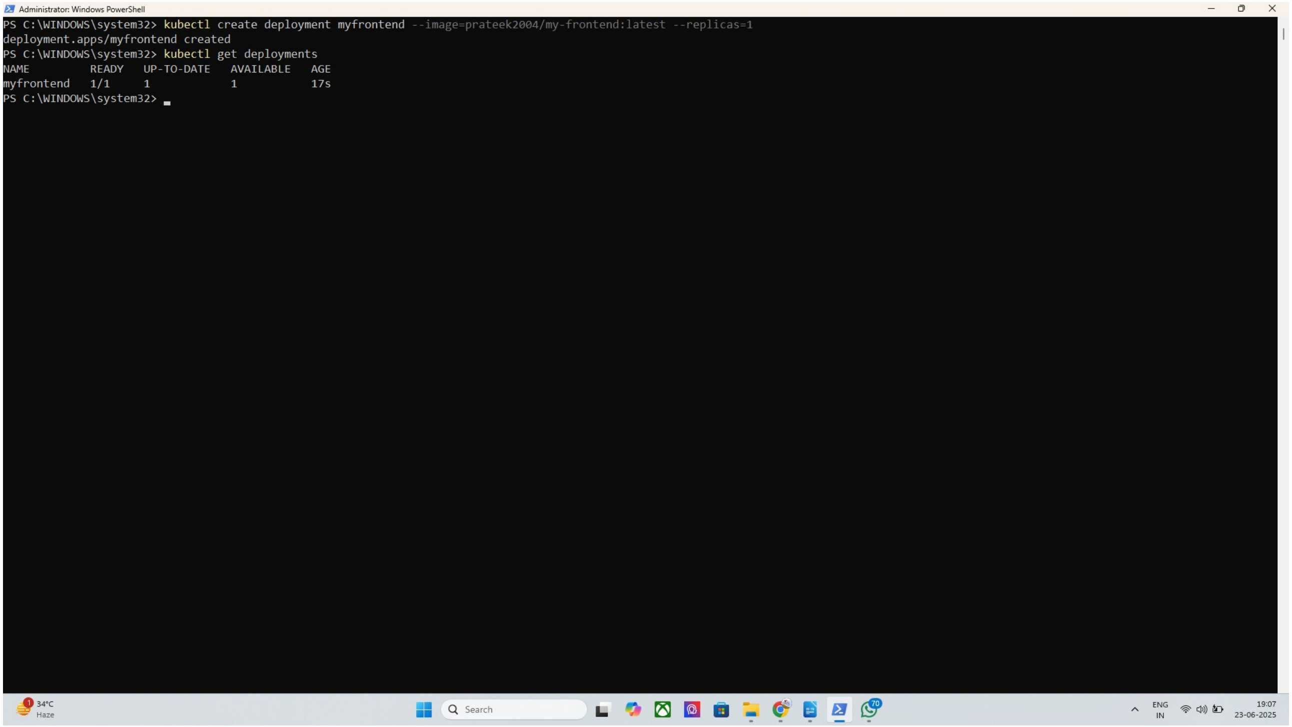
COMMAND USED : kubectl get deployments
kubectl get pods

Explanation (for your DevOps project report)

1. `kubectl get deployments`
   ➤ Lists all Kubernetes deployments in the current namespace (defaults to `default`).
   📌 Output: *"No resources found in default namespace"* — means no deployments exist there yet.
2. `kubectl get pods`
   ➤ Lists all pods in the default namespace.
   📌 Output: *"No resources found"* — confirms no running workloads currently exist in `default`.



```
PS C:\WINDOWS\system32> kubectl get deployments
No resources found in default namespace.
PS C:\WINDOWS\system32> kubectl get pods
No resources found in default namespace.
PS C:\WINDOWS\system32>
```



```
PS C:\WINDOWS\system32> kubectl create deployment myfrontend --image=prateek2004/my-frontend:latest --replicas=1
deployment.apps/myfrontend created
PS C:\WINDOWS\system32> kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
myfrontend   1/1     1            1           17s
PS C:\WINDOWS\system32> _
```

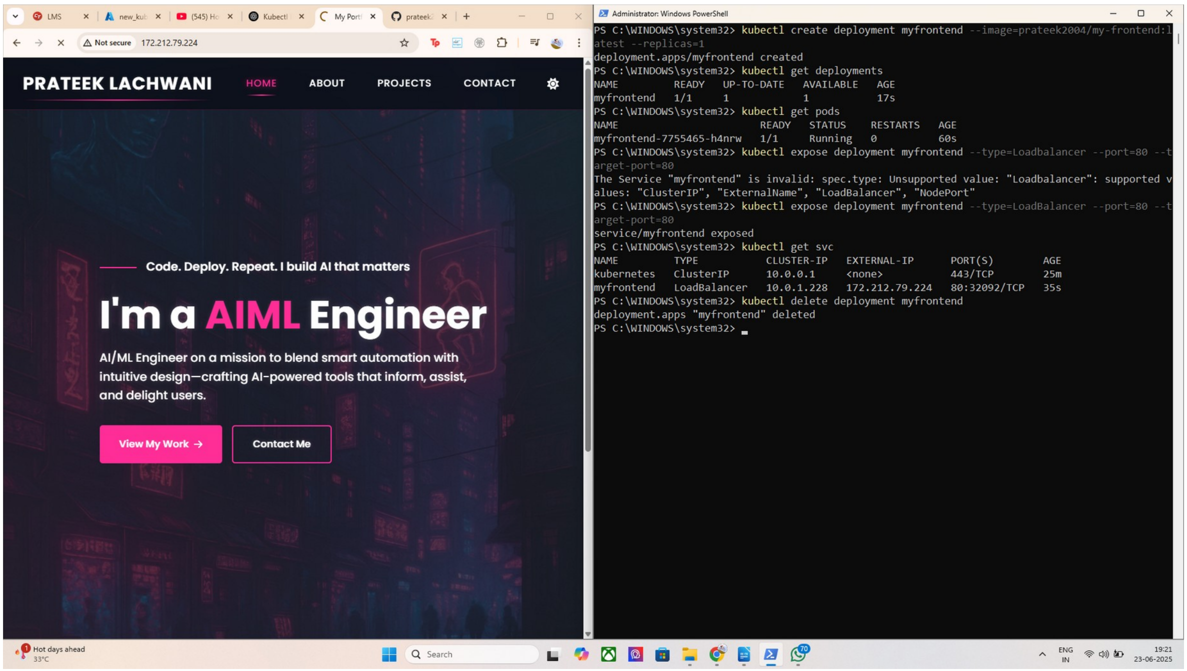
LETS CREATE A DEPLOYMENT AROUND MY FRONTEND

COMMAND USED :

kubectl create deployment myfrontend --image=prateek2004/my-frontend:latest --replicas=1
kubectl get deployments
kubectl get pods
kubectl expose deployment myfrontend --type=LoadBalancer --port=80 --target-port=80
kubectl get svc

Explanation (3–4 lines for your report)

A custom frontend container was deployed to the AKS cluster using `kubectl create deployment` with a public Docker image.
It was then exposed via a `LoadBalancer` service to make it accessible externally.
The external IP **`172.212.79.224`** allowed public access to the app.



Code. Deploy. Repeat. I build AI that matters

I'm a AIML Engineer

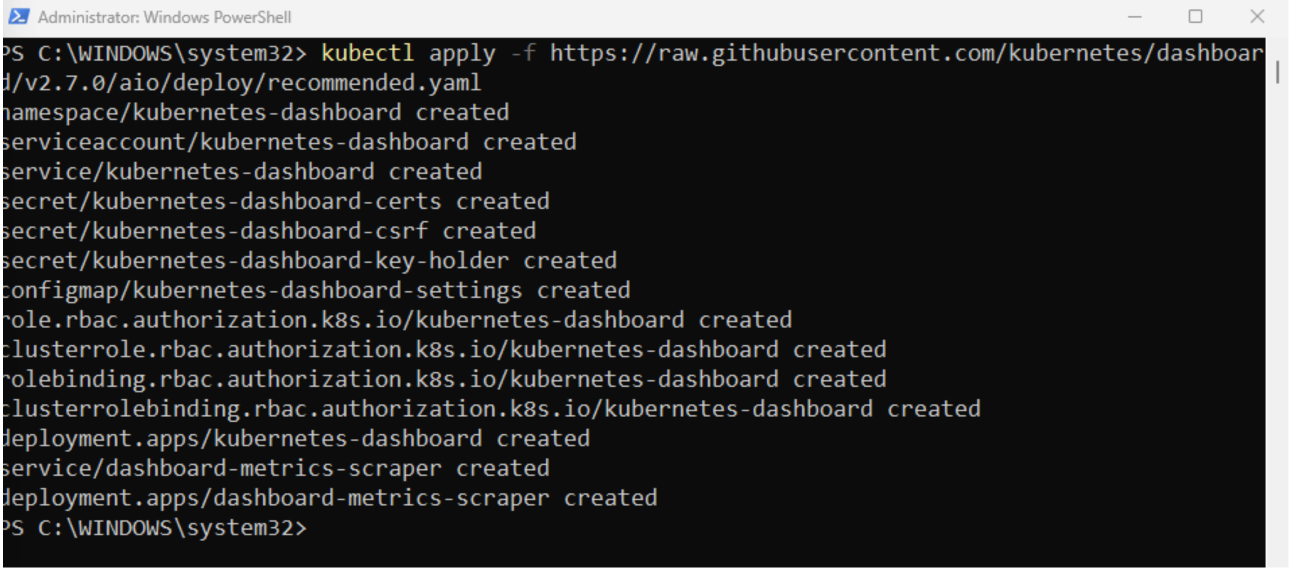
AI/ML Engineer on a mission to blend smart automation with intuitive design—crafting AI-powered tools that inform, assist, and delight users.

```
PS C:\WINDOWS\system32> kubectl create deployment myfrontend --image=prateek2004/my-frontend:l
atest --replicas=1
deployment.apps/myfrontend created
PS C:\WINDOWS\system32> kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
myfrontend   1/1     1            1           17s
PS C:\WINDOWS\system32> kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
myfrontend-7755465-h4nrw   1/1     Running   0          60s
PS C:\WINDOWS\system32> kubectl expose deployment myfrontend --type=Loadbalancer --port=80 --t
arget-port=80
The Service "myfrontend" is invalid: spec.type: Unsupported value: "Loadbalancer": supported v
alues: "ClusterIP", "ExternalName", "LoadBalancer", "NodePort"
PS C:\WINDOWS\system32> kubectl expose deployment myfrontend --type=LoadBalancer --port=80 --t
arget-port=80
service/myfrontend exposed
PS C:\WINDOWS\system32> kubectl get svc
NAME         TYPE           CLUSTER-IP   EXTERNAL-IP      PORT(S)        AGE
kubernetes   ClusterIP      10.0.0.1     <none>           443/TCP        25m
myfrontend   LoadBalancer   10.0.1.228   172.212.79.224   80:32092/TCP   35s
PS C:\WINDOWS\system32> kubectl delete deployment myfrontend
deployment.apps "myfrontend" deleted
PS C:\WINDOWS\system32> _
```

CONFIGURE THE DASHBOARD TO THE TERMINAL

COMMAND USED :

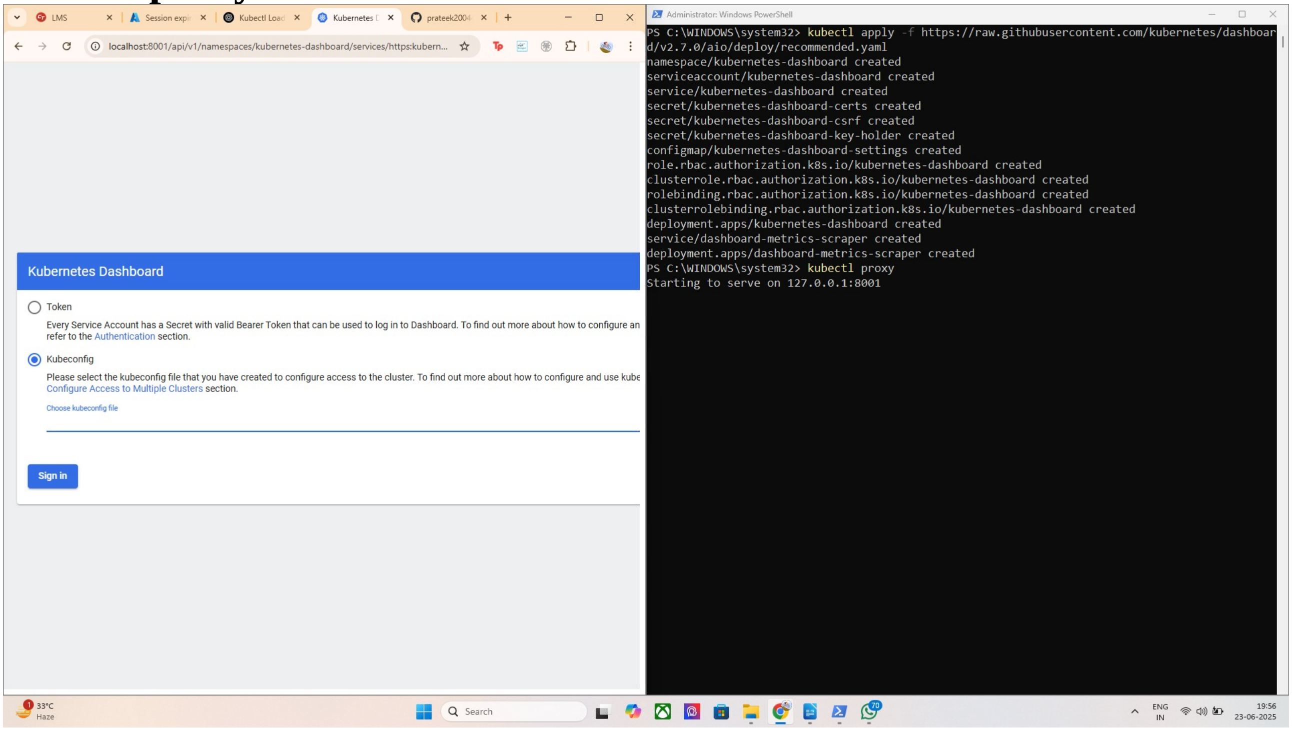
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml



```
PS C:\WINDOWS\system32> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboar
d/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
PS C:\WINDOWS\system32>
```

command to access the dashboard

kubectl proxy



LMS
Session expir
Kubectl Load
Kubernetes D
prateek2004
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubern...
Administrator: Windows PowerShell
33°C
Haze
Search
ENG
IN
19:56
23-06-2025

Kubernetes Dashboard

○ Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure an
refer to the Authentication section.

● Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kube
Configure Access to Multiple Clusters section.

Choose kubeconfig file

Sign in

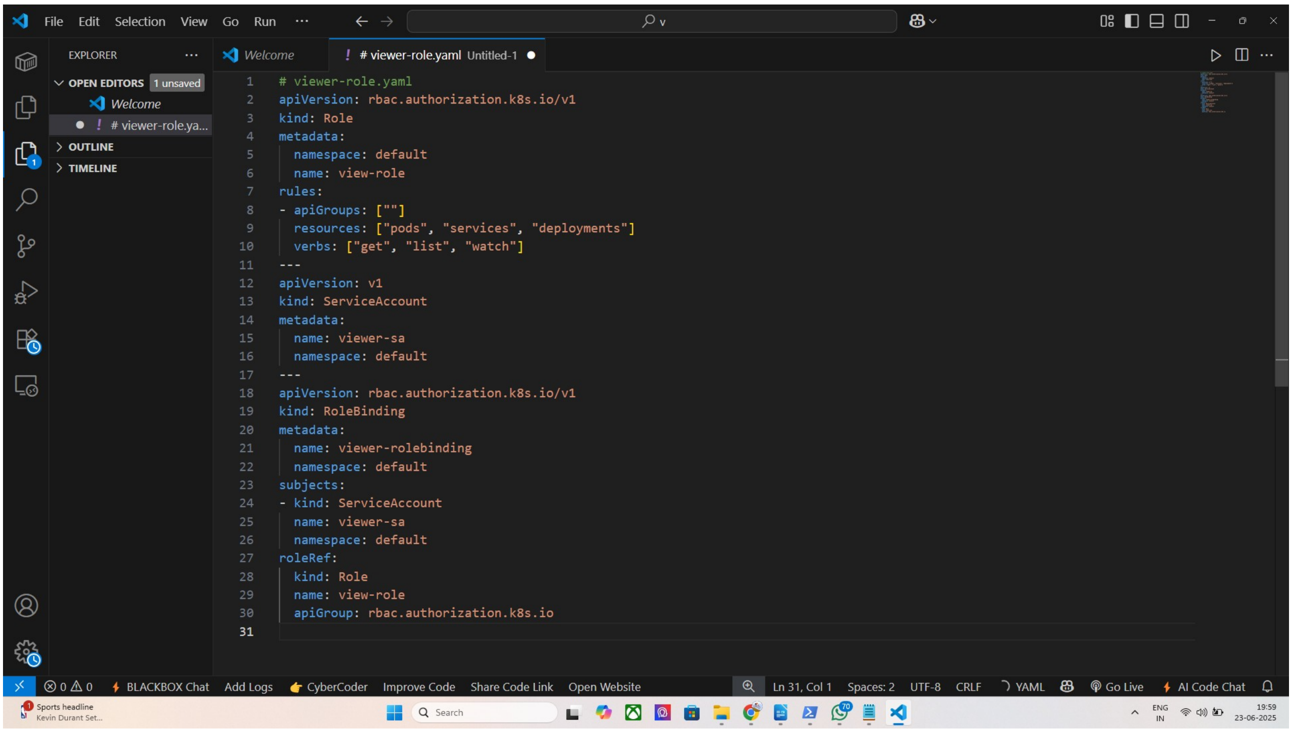
```
PS C:\WINDOWS\system32> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboar
d/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
PS C:\WINDOWS\system32> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

PART B

Lets Create Service Accounts & Roles for Users

Create YAMLs for the following:

a. Viewer (Read-only access):


File Edit Selection View Go Run ···
EXPLORER
OPEN EDITORS 1 unsaved
Welcome
! # viewer-role.ya...
OUTLINE
TIMELINE
Welcome
! # viewer-role.yaml Untitled-1 ●
⊗ 0 ⚠ 0 BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Open Website
Ln 31, Col 1 Spaces: 2 UTF-8 CRLF YAML Go Live AI Code Chat
Sports headline
Kevin Durant Set...
Search
ENG IN
19:59
23-06-2025

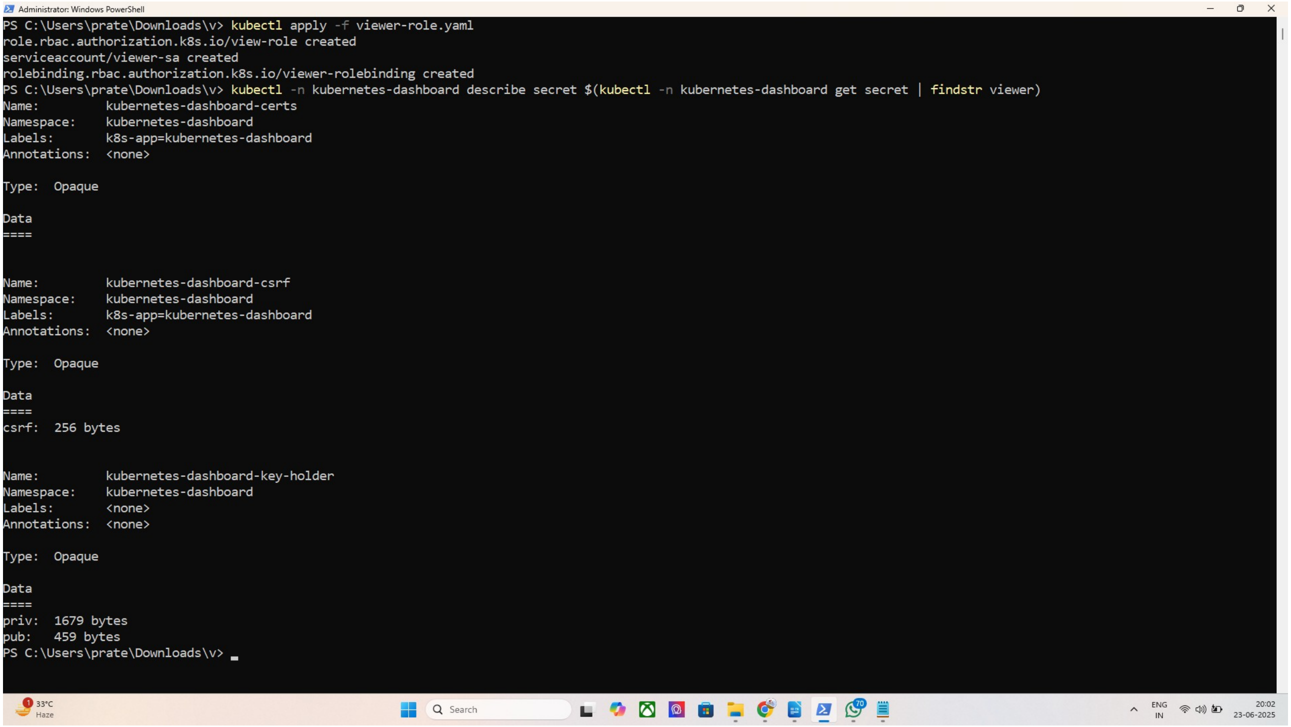
```
# viewer-role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: default
  name: view-role
rules:
- apiGroups: [""]
  resources: ["pods", "services", "deployments"]
  verbs: ["get", "list", "watch"]
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: viewer-sa
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: viewer-rolebinding
  namespace: default
subjects:
- kind: ServiceAccount
  name: viewer-sa
  namespace: default
roleRef:
  kind: Role
  name: view-role
  apiGroup: rbac.authorization.k8s.io
```

Save it as veiwer_role.yaml

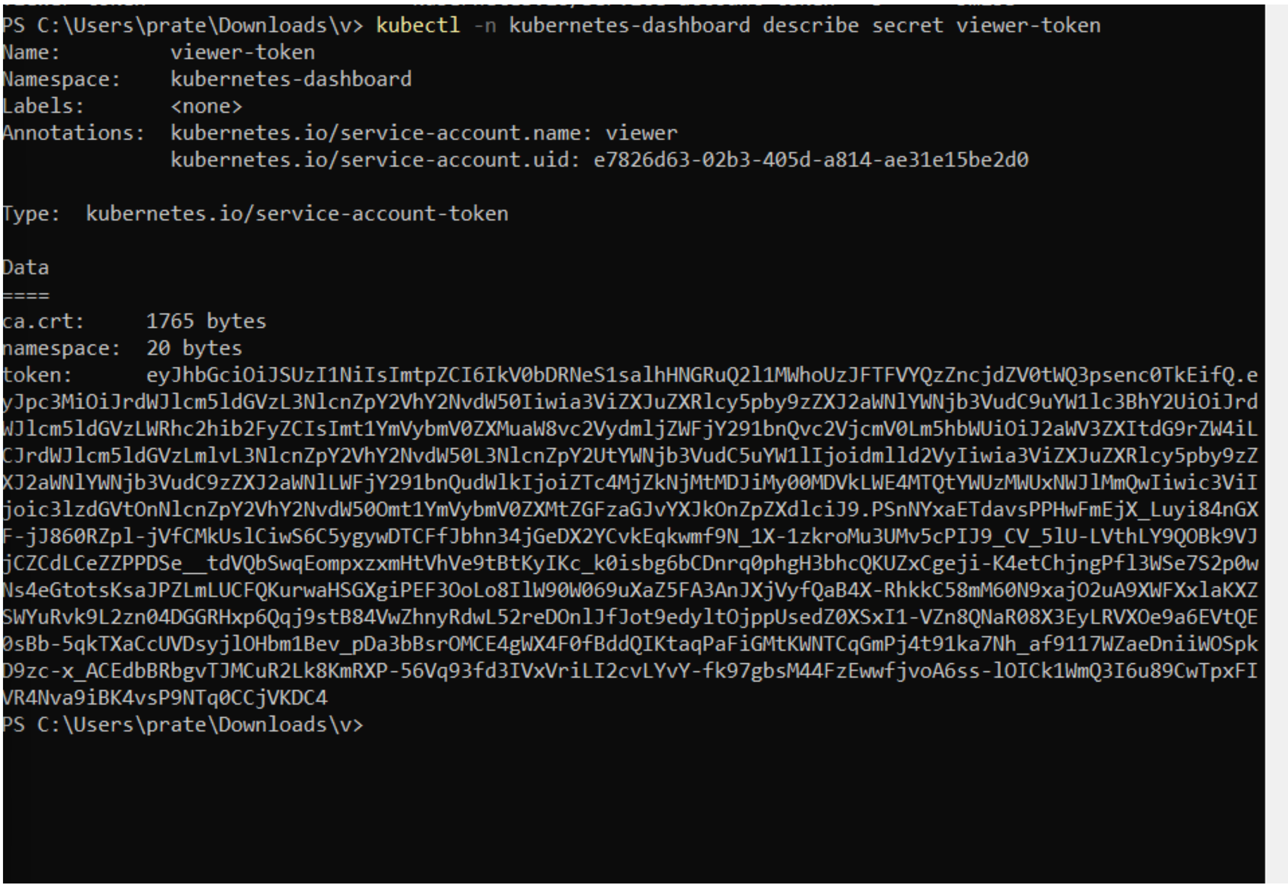
and in the terminal change directory to the created path
and run command : kubectl apply -f viewer-access.yaml

Get the Token to Log In

command used : kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | findstr viewer)

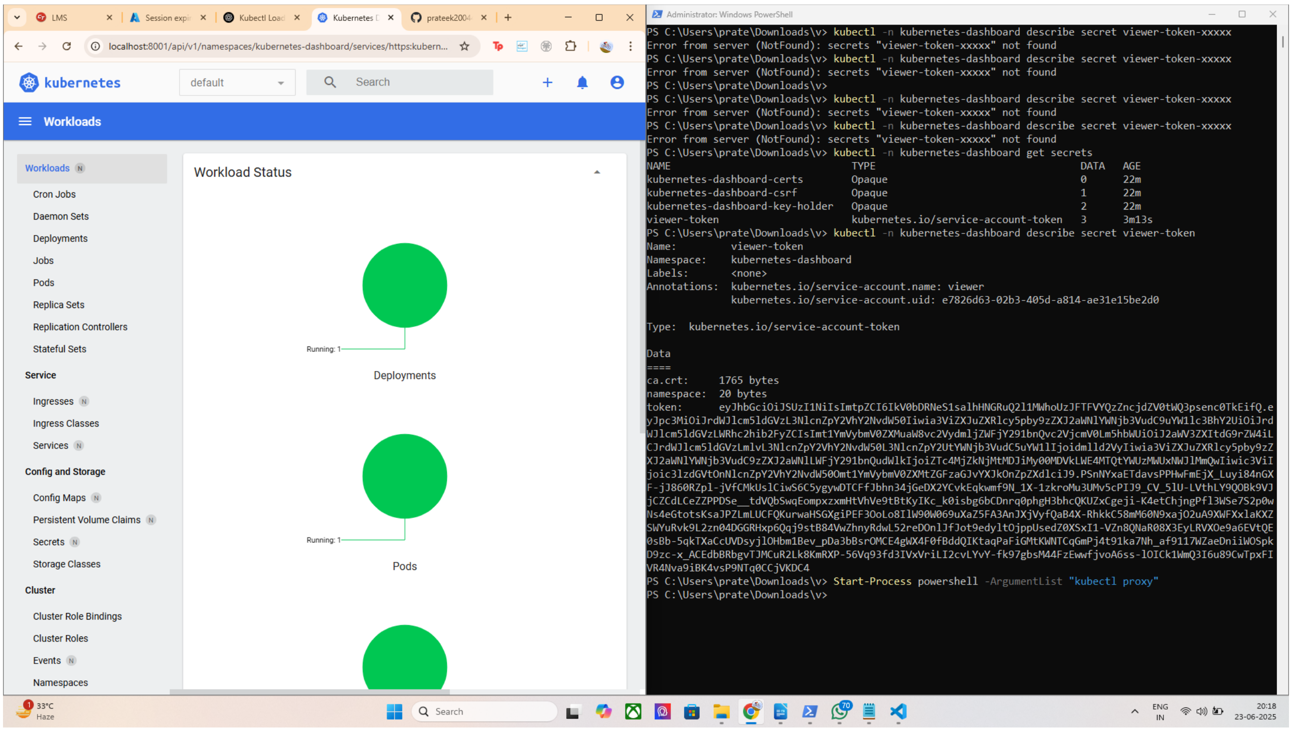


```
PS C:\Users\prate\Downloads\v> kubectl apply -f viewer-role.yaml
role.rbac.authorization.k8s.io/view-role created
serviceaccount/viewer-sa created
rolebinding.rbac.authorization.k8s.io/viewer-rolebinding created
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | findstr viewer)
Name:         kubernetes-dashboard-certs
Namespace:    kubernetes-dashboard
Labels:       k8s-app=kubernetes-dashboard
Annotations:  <none>

Type:  Opaque

Data
====


Name:         kubernetes-dashboard-csrf
Namespace:    kubernetes-dashboard
Labels:       k8s-app=kubernetes-dashboard
Annotations:  <none>

Type:  Opaque

Data
====
csrf:  256 bytes


Name:         kubernetes-dashboard-key-holder
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
priv:  1679 bytes
pub:   459 bytes
PS C:\Users\prate\Downloads\v>
```

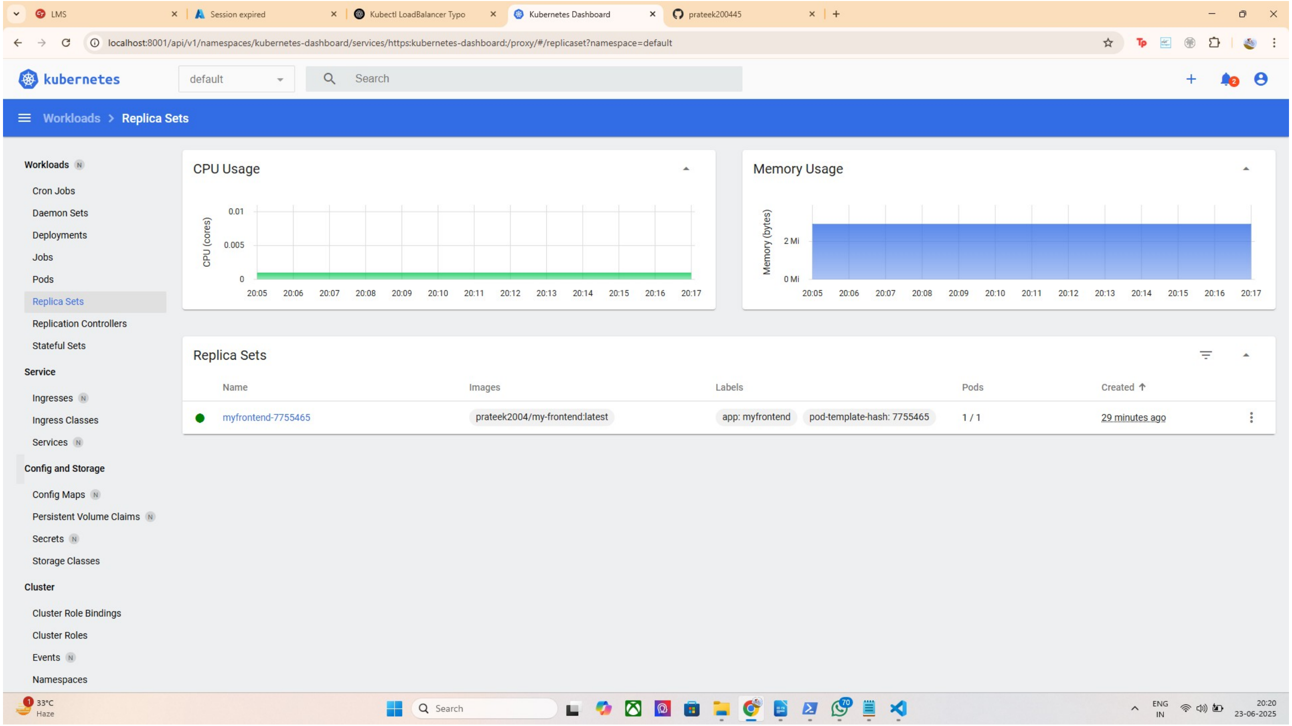


```
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token
Name:         viewer-token
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: viewer
              kubernetes.io/service-account.uid: e7826d63-02b3-405d-a814-ae31e15be2d0

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1765 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkV0bDRNeS1salhHNGRuQ2l1MWhoUzJFTFVYQzZncjdZV0tWQ3psenc0TkEifQ.e
yJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrd
WJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJ2aWV3ZXItdG9rZW4iL
CJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoidmlld2VyIiwia3ViZXJuZXRlcy5pby9zZ
XJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZTc4MjZkNjMtMDJiMy00MDVkLWE4MTQtYWUzMWUxNWJlMmQwIiwic3ViI
joic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOnZpZXdlciJ9.PSnNYxaETdavsPPHwFmEjX_Luyi84nGX
F-jJ860RZpl-jVfCMkUslCiwS6C5ygywDTCFfJbhn34jGeDX2YCvkEqkwmf9N_1X-1zkroMu3UMv5cPIJ9_CV_5lU-LVthLY9QOBk9VJ
jCZCdLCeZZPPDSe__tdVQbSwqEompxzxmHtVhVe9tBtKyIKc_k0isbg6bCDnrq0phgH3bhcQKUZxCgeji-K4etChjngPfl3WSe7S2p0w
Ns4eGtotsKsaJPZLmLUCFQKurwaHSGXgiPEF3OoLo8IlW90W069uXaZ5FA3AnJXjVyfQaB4X-RhkkC58mM60N9xajO2uA9XWFXxlaKXZ
SWYuRvk9L2zn04DGGRHxp6Qqj9stB84VwZhnyRdwL52reDOnlJfJot9edyltOjppUsedZ0XSxI1-VZn8QNaR08X3EyLRVXOe9a6EVtQE
0sBb-5qkTXaCcUVDsyjlOHbm1Bev_pDa3bBsrOMCE4gWX4F0fBddQIKtaqPaFiGMtKWNTCqGmPj4t91ka7Nh_af9117WZaeDniiWOSpk
D9zc-x_ACEdbBRbgvTJMCuR2Lk8KmRXP-56Vq93fd3IVxVriLI2cvLYvY-fk97gbsM44FzEwwfjvoA6ss-lOICk1WmQ3I6u89CwTpxFI
VR4Nva9iBK4vsP9NTq0CCjVKDC4
PS C:\Users\prate\Downloads\v>
```

DASHBOARD SUCCESSFULLY ACCESSED USING ONLY VEIWER ROLE



```
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token-xxxxx
Error from server (NotFound): secrets "viewer-token-xxxxx" not found
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token-xxxxx
Error from server (NotFound): secrets "viewer-token-xxxxx" not found
PS C:\Users\prate\Downloads\v>
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token-xxxxx
Error from server (NotFound): secrets "viewer-token-xxxxx" not found
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token-xxxxx
Error from server (NotFound): secrets "viewer-token-xxxxx" not found
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard get secrets
NAME                              TYPE                                  DATA   AGE
kubernetes-dashboard-certs        Opaque                                0      22m
kubernetes-dashboard-csrf         Opaque                                1      22m
kubernetes-dashboard-key-holder   Opaque                                2      22m
viewer-token                      kubernetes.io/service-account-token   3      3m13s
PS C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret viewer-token
Name:         viewer-token
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: viewer
              kubernetes.io/service-account.uid: e7826d63-02b3-405d-a814-ae31e15be2d0

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1765 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkV0bDRNeS1salhHNGRuQ2l1MWhoUzJFTFVYQzZncjdZV0tWQ3psenc0TkEifQ.e
yJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrd
WJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJ2aWV3ZXItdG9rZW4iL
CJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoidmlld2VyIiwia3ViZXJuZXRlcy5pby9zZ
XJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZTc4MjZkNjMtMDJiMy00MDVkLWE4MTQtYWUzMWUxNWJlMmQwIiwic3ViI
joic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOnZpZXdlciJ9.PSnNYxaETdavsPPHwFmEjX_Luyi84nGX
F-jJ860RZpl-jVfCMkUslCiwS6C5ygywDTCFfJbhn34jGeDX2YCvkEqkwmf9N_1X-1zkroMu3UMv5cPIJ9_CV_5lU-LVthLY9QOBk9VJ
jCZCdLCeZZPPDSe__tdVQbSwqEompxzxmHtVhVe9tBtKyIKc_k0isbg6bCDnrq0phgH3bhcQKUZxCgeji-K4etChjngPfl3WSe7S2p0w
Ns4eGtotsKsaJPZLmLUCFQKurwaHSGXgiPEF3OoLo8IlW90W069uXaZ5FA3AnJXjVyfQaB4X-RhkkC58mM60N9xajO2uA9XWFXxlaKXZ
SWYuRvk9L2zn04DGGRHxp6Qqj9stB84VwZhnyRdwL52reDOnlJfJot9edyltOjppUsedZ0XSxI1-VZn8QNaR08X3EyLRVXOe9a6EVtQE
0sBb-5qkTXaCcUVDsyjlOHbm1Bev_pDa3bBsrOMCE4gWX4F0fBddQIKtaqPaFiGMtKWNTCqGmPj4t91ka7Nh_af9117WZaeDniiWOSpk
D9zc-x_ACEdbBRbgvTJMCuR2Lk8KmRXP-56Vq93fd3IVxVriLI2cvLYvY-fk97gbsM44FzEwwfjvoA6ss-lOICk1WmQ3I6u89CwTpxFI
VR4Nva9iBK4vsP9NTq0CCjVKDC4
PS C:\Users\prate\Downloads\v> Start-Process powershell -ArgumentList "kubectl proxy"
PS C:\Users\prate\Downloads\v>
```



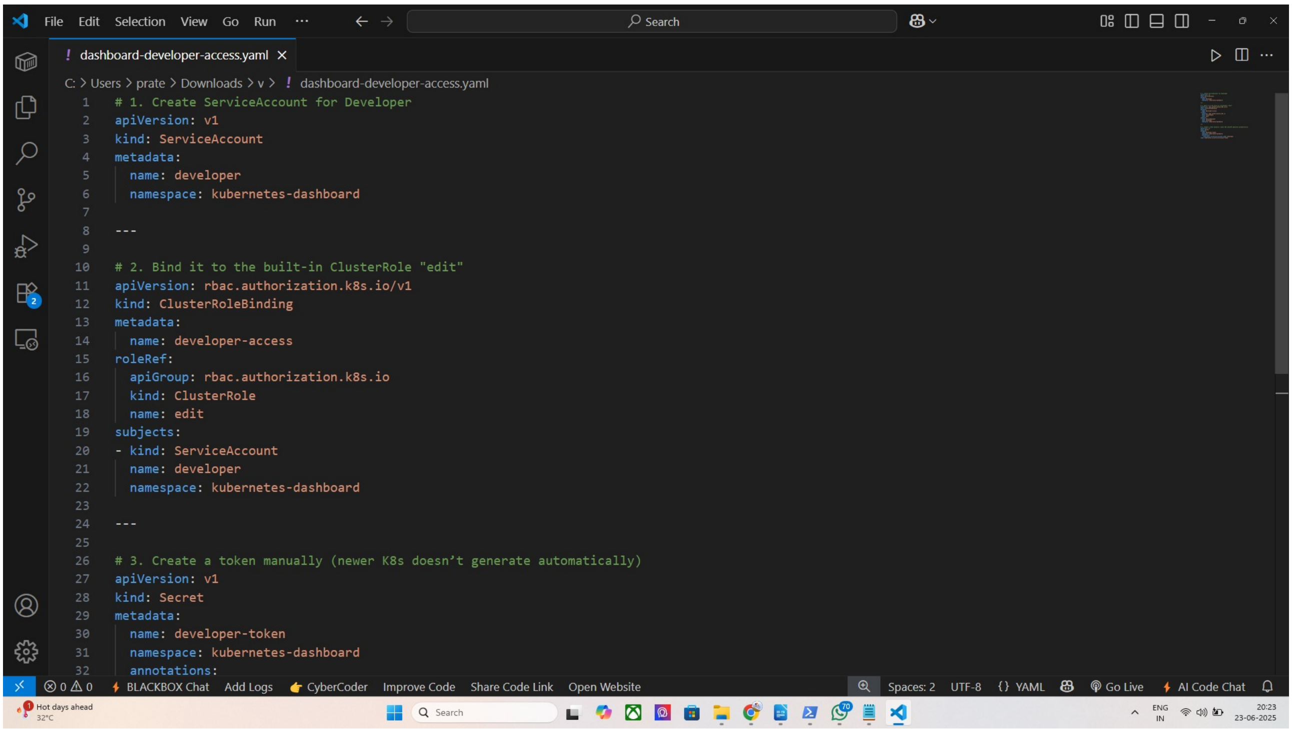
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/replicaset?namespace=default

Replica Sets

Name	Images	Labels	Pods	Created ↑
● myfrontend-7755465	prateek2004/my-frontend:latest	app: myfrontend pod-template-hash: 7755465	1 / 1	29 minutes ago

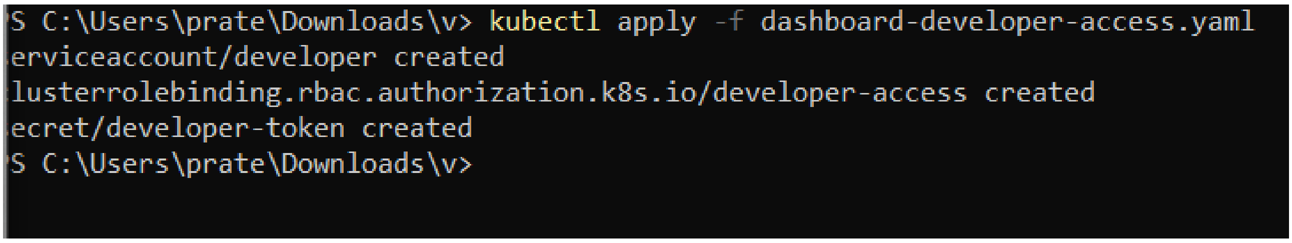
LETS CREATE DASHBOARD FOR DEVELOPER

Step-by-Step YAML: `dashboard-developer-access.yaml`

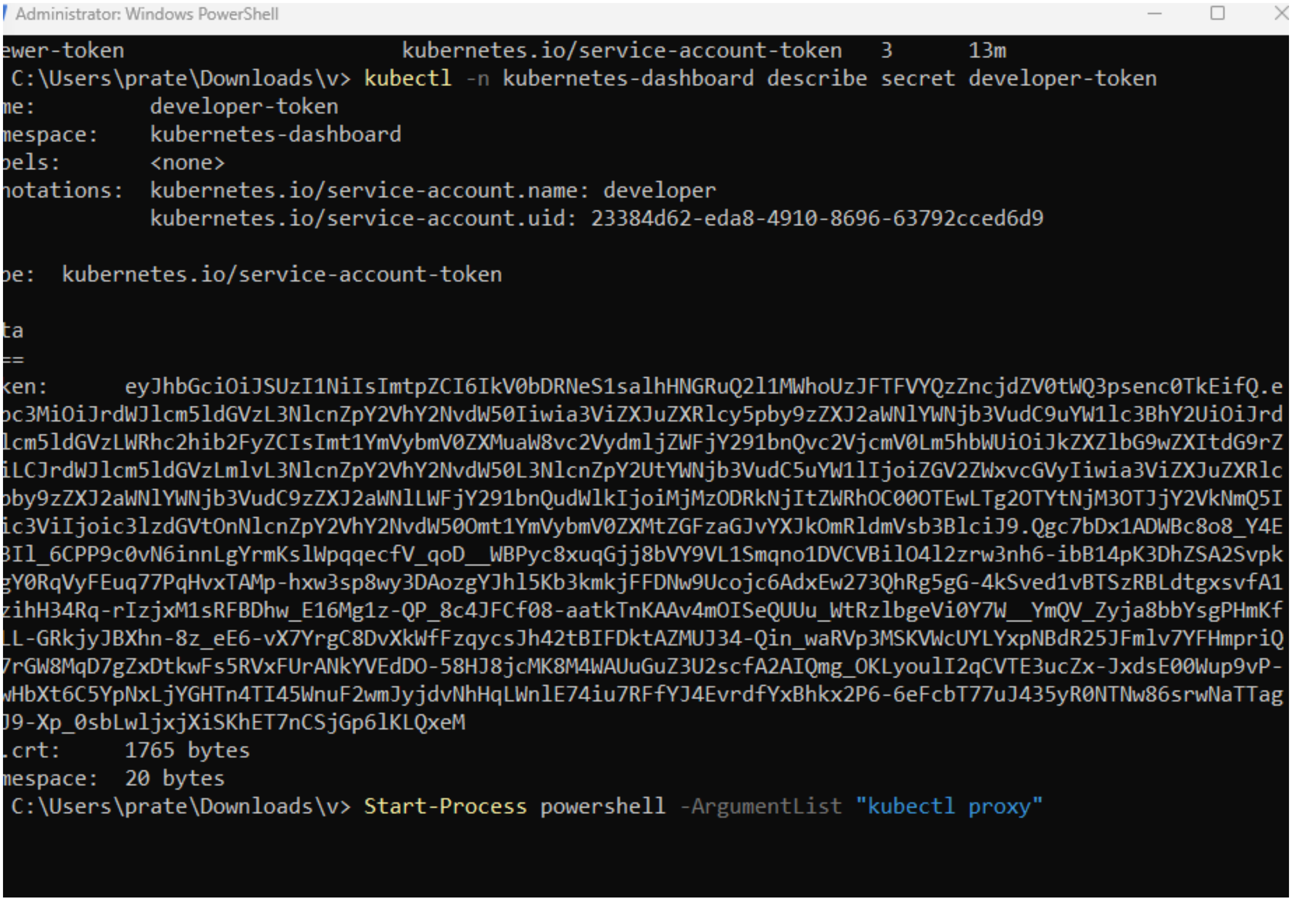


File Edit Selection View Go Run ···
Search
dashboard-developer-access.yaml
C: > Users > prate > Downloads > v > ! dashboard-developer-access.yaml
BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Open Website
Spaces: 2 UTF-8 {} YAML Go Live AI Code Chat

```yaml
# 1. Create ServiceAccount for Developer
apiVersion: v1
kind: ServiceAccount
metadata:
  name: developer
  namespace: kubernetes-dashboard

---

# 2. Bind it to the built-in ClusterRole "edit"
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: developer-access
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: edit
subjects:
- kind: ServiceAccount
  name: developer
  namespace: kubernetes-dashboard

---

# 3. Create a token manually (newer K8s doesn’t generate automatically)
apiVersion: v1
kind: Secret
metadata:
  name: developer-token
  namespace: kubernetes-dashboard
  annotations:
```

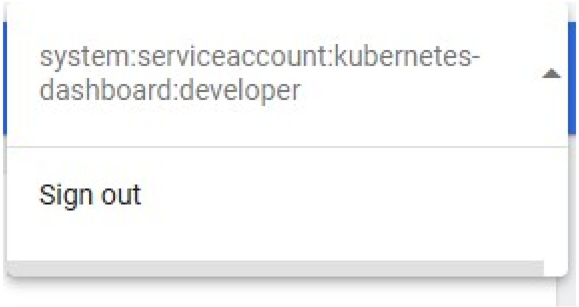


```
S C:\Users\prate\Downloads\v> kubectl apply -f dashboard-developer-access.yaml
erviceaccount/developer created
lusterrolebinding.rbac.authorization.k8s.io/developer-access created
ecret/developer-token created
S C:\Users\prate\Downloads\v>
```



```
ewer-token                     kubernetes.io/service-account-token   3      13m
 C:\Users\prate\Downloads\v> kubectl -n kubernetes-dashboard describe secret developer-token
me:         developer-token
mespace:    kubernetes-dashboard
bels:       <none>
notations:  kubernetes.io/service-account.name: developer
            kubernetes.io/service-account.uid: 23384d62-eda8-4910-8696-63792cced6d9

pe:  kubernetes.io/service-account-token

ta
==
ken:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkV0bDRNeS1salhHNGRuQ2l1MWhoUzJFTFVYQzZncjdZV0tWQ3psenc0TkEifQ.e
pc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrd
lcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJkZXZlbG9wZXItdG9rZ
iLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoiZGV2ZWxvcGVyIiwia3ViZXJuZXRlc
pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiMjMzODRkNjItZWRhOC00OTEwLTg2OTYtNjM3OTJjY2VkNmQ5I
ic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOmRldmVsb3BlciJ9.Qgc7bDx1ADWBc8o8_Y4E
3Il_6CPP9c0vN6innLgYrmKslWpqqecfV_qoD__WBPyc8xuqGjj8bVY9VL1Smqno1DVCVBilO4l2zrw3nh6-ibB14pK3DhZSA2Svpk
gY0RqVyFEuq77PqHvxTAMp-hxw3sp8wy3DAozgYJhl5Kb3kmkjFFDNw9Ucojc6AdxEw273QhRg5gG-4kSved1vBTSzRBLdtgxsvfA1
zihH34Rq-rIzjxM1sRFBDhw_E16Mg1z-QP_8c4JFCf08-aatkTnKAAv4mOISeQUUu_WtRzlbgeVi0Y7W__YmQV_Zyja8bbYsgPHmKf
LL-GRkjyJBXhn-8z_eE6-vX7YrgC8DvXkWfFzqycsJh42tBIFDktAZMUJ34-Qin_waRVp3MSKVWcUYLYxpNBdR25JFmlv7YFHmpriQ
7rGW8MqD7gZxDtkwFs5RVxFUrANkYVEdDO-58HJ8jcMK8M4WAUuGuZ3U2scfA2AIQmg_OKLyoulI2qCVTE3ucZx-JxdsE00Wup9vP-
wHbXt6C5YpNxLjYGHTn4TI45WnuF2wmJyjdvNhHqLWnlE74iu7RFfYJ4EvrdfYxBhkx2P6-6eFcbT77uJ435yR0NTNw86srwNaTTag
J9-Xp_0sbLwljxjXiSKhET7nCSjGp6lKLQxeM
.crt:     1765 bytes
mespace:  20 bytes
 C:\Users\prate\Downloads\v> Start-Process powershell -ArgumentList "kubectl proxy"
```

http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/



TO LEARN MORE ABOUT Real-world use case for **multi-user RBAC in AKS**.

Deploy a sample app (like nginx or your own frontend) and restrict who can see it.

Step 1: Deploy the `nginx` App in a Separate Namespace

Create `dev` namespace:

```
kubectl create namespace dev
```

Deploy `OWNFRONTEND/NGINX` in that namespace:

```
kubectl create deployment nginx --image=nginx --namespace=dev
```

You can verify:

```
kubectl get pods -n dev
```

Step 2: Restrict Access to `dev` Namespace

By default, users with the `view` or `edit` **ClusterRoleBinding** have **access to all namespaces**.

So instead, we'll:

- **Remove or not use ClusterRoleBinding**
- Use **Role** + **RoleBinding** in `dev` namespace only

Step 3: Configure RBAC to Restrict Access

Create role access only for `developer` in `dev` namespace

📄 `dev-namespace-developer-access.yaml`:

```
# 1. Role with edit permissions in "dev" namespace
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: dev
  name: developer-edit
rules:
- apiGroups: ["", "apps", "extensions"]
  resources: ["pods", "services", "deployments", "replicasets"]
  verbs: ["get", "list", "create", "update", "delete", "watch"]
```

```
---

# 2. Bind developer SA to this Role
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: developer-edit-binding
  namespace: dev
subjects:
- kind: ServiceAccount
  name: developer
  namespace: kubernetes-dashboard
roleRef:
  kind: Role
  name: developer-edit
  apiGroup: rbac.authorization.k8s.io
```

Step 4: Restrict `viewer` From `dev` Namespace

If you used a `ClusterRoleBinding` for `viewer`, it already has access to all namespaces.

To restrict it:

1. **Delete that binding**:

   ```
   kubectl delete clusterrolebinding viewer-access
   ```

2. Recreate access **only for `default` namespace**:

`viewer-default-access.yaml`

```
yaml

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: viewer-read
  namespace: default
rules:
- apiGroups: [""]
  resources: ["pods", "services", "deployments"]
  verbs: ["get", "list", "watch"]

---

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: viewer-read-binding
  namespace: default
subjects:
- kind: ServiceAccount
  name: viewer
  namespace: kubernetes-dashboard
roleRef:
  kind: Role
  name: viewer-read
```

```
  apiGroup: rbac.authorization.k8s.io
```

Apply it:

```
kubectl apply -f viewer-default-access.yaml
```

Final Behavior:

User	Namespace Access	Permissions
viewer	`default` only	Read-only
developer	`dev` only	Full app management

HENCE ITS SUCCESSFULLY VERIFIED

- Log in as **viewer** → you'll see only `default` resources
- Log in as **developer** → you can manage `nginx/MYFRONTEND`Deploy an AKS cluster using the portal. Access the dashboard and create roles for multiple users

I successfully deployed and exposed a microservice application on AKS using all three service types, completing both:
`http://172.212.79.224`

- ✅ **Task 4: Microservice deployment on AKS with public access**
- ✅ **Task 5: Service exposure using ClusterIP, NodePort, and LoadBalancer**

Objective:

To deploy a frontend microservice on Azure Kubernetes Service (AKS) and expose it using all three Kubernetes service types — **ClusterIP**, **NodePort**, and **LoadBalancer**. This validates application hosting, access control, and cloud-native service exposure.

Steps I Followed to Complete the Task:

Step 1: Set Up Access to AKS Cluster

I first connected my terminal to my AKS cluster named `new_kuber` using Azure CLI:

```
az configure --defaults group=prateek_cluster
az aks get-credentials --name new_kuber
kubectl config current-context
```

Then I verified the nodes were running:

```
kubectl get nodes
```

Step 2: Deploy a Microservice (Custom Frontend)

I used my custom Docker image hosted on Docker Hub (`prateek2004/my-frontend:latest`) to deploy a microservice with 1 replica:

```
kubectl create deployment myfrontend --
image=prateek2004/my-frontend:latest --replicas=1
```

To verify the deployment:

```
kubectl get deployments
kubectl get pods
```

Step 3: Expose the App via LoadBalancer (Public Internet)

To access the application publicly, I exposed it using a **LoadBalancer** service:

```
kubectl expose deployment myfrontend --
type=LoadBalancer --port=80 --target-port=80
kubectl get svc
```

Output:

```
NAME          TYPE             CLUSTER-IP
EXTERNAL-IP      PORT(S)
myfrontend    LoadBalancer     10.0.X.X
172.212.79.224   80:32XXX/TCP
```

I then opened the public IP `http://172.212.79.224` in the browser and successfully viewed my personal portfolio web page.

Step 4: Expose the App via ClusterIP (Default Service)

For internal-only service communication (used inside the cluster), I exposed the same deployment as a **ClusterIP**:

```
kubectl expose deployment myfrontend --
type=ClusterIP --port=80 --target-port=80
```

ClusterIP is the **default** service type and allows internal traffic routing only.

Step 5: Expose the App via NodePort (Node-Level Port Mapping)

To simulate edge routing without cloud load balancing, I exposed the app via **NodePort**:

```
kubectl expose deployment myfrontend --
type=NodePort --port=80 --target-port=80
kubectl get svc
```

Sample Output:

```
myfrontend   NodePort   10.0.X.X   <none>
80:30036/TCP
```

Though direct node IP access may be limited in AKS, this demonstrates how services can be mapped to static ports on every node.