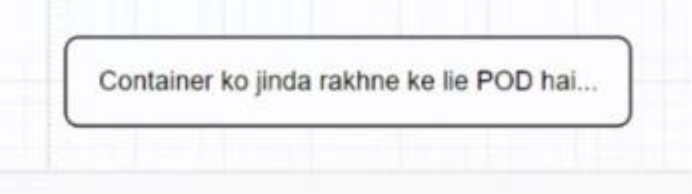
02 November 2024

1)

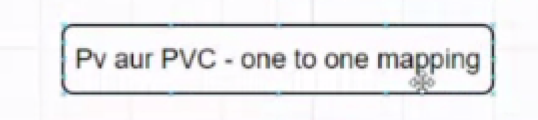


Container ko jinda rakhne ke lie POD hai...

2) Storage class is of 2 types – static and dynamic

+++

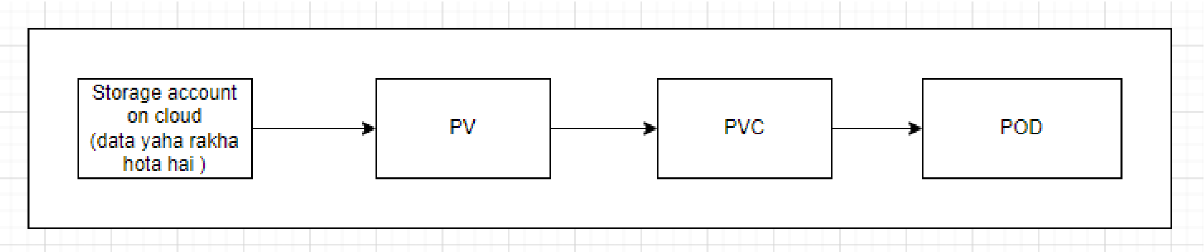
AGENDA = DYNAMIC STORAGE CLASS



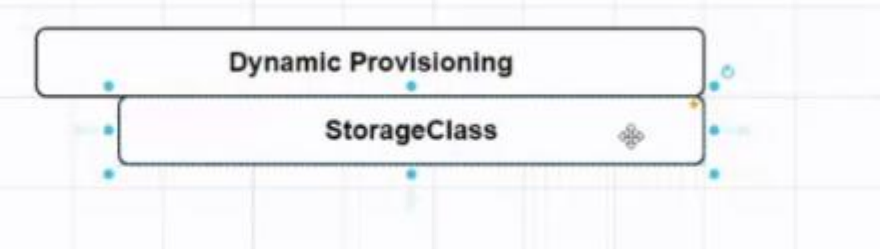
Pv aur PVC - one to one mapping

21) PV and pvc me 1 to 1 mapping hoti hai mtlb 1 pv se 1 hi pvc connect hoga. But 1 pvc can be mounted on multiple pods. Basically pvc ek hi rhega.

22) Dynamic provisioning = Ek aisa tarika jisme pvc banne ke baad pv aur storage account khud se ban jaata hai



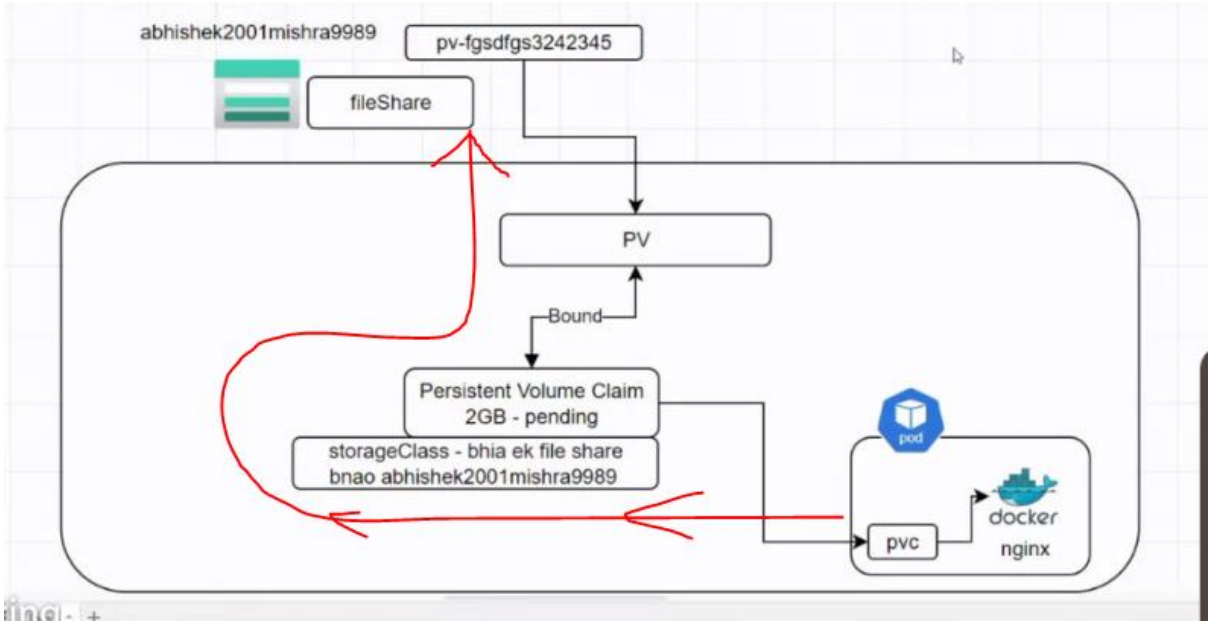
23) Dynamic provisioning of volume in k8s is done by “Storage class”.



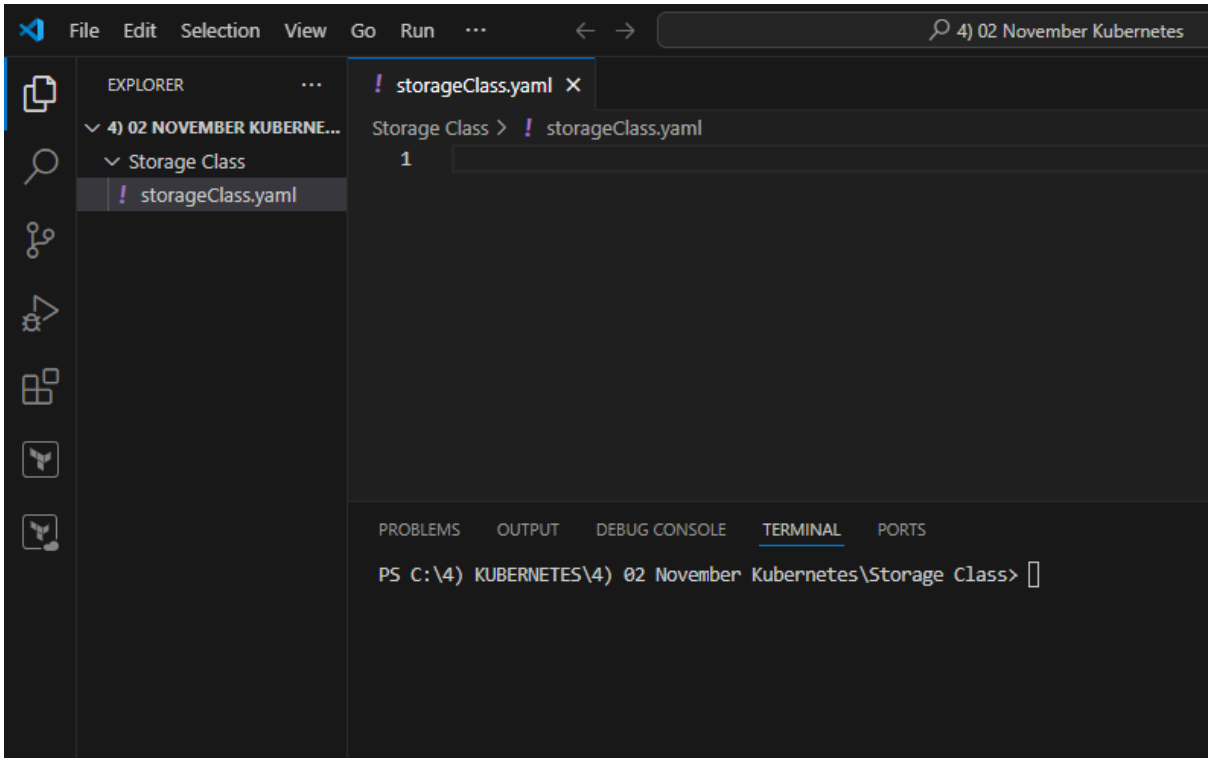
Dynamic Provisioning

StorageClass

24) Storage class actually automates the process of making file share and persistent volume.



25) Now making storage class so create folder “4) 02 November Kubernetes” and in that create Storage class folder and file in vs code



26) Connect k8s cluster in it

az login

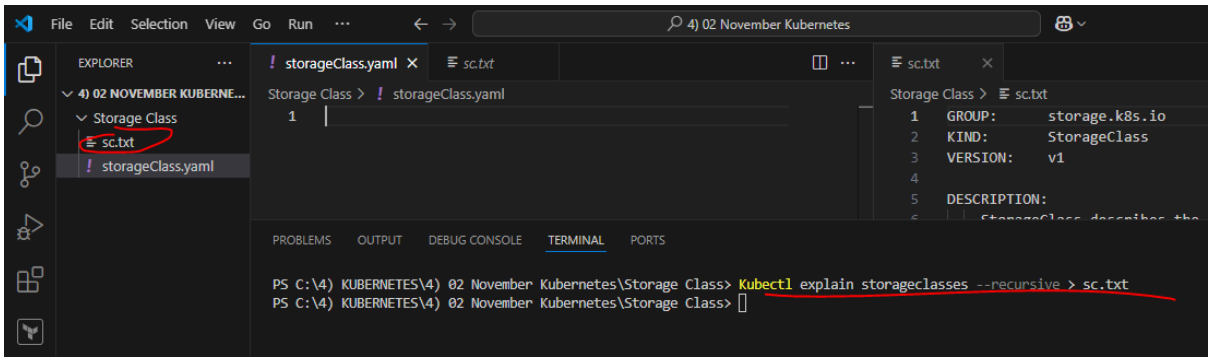
az account set --subscription 48f88df7-0d53-4866-a66f-82eb0ac469e3

az aks get-credentials --resource-group rgtoday --name k8today --overwrite-existing

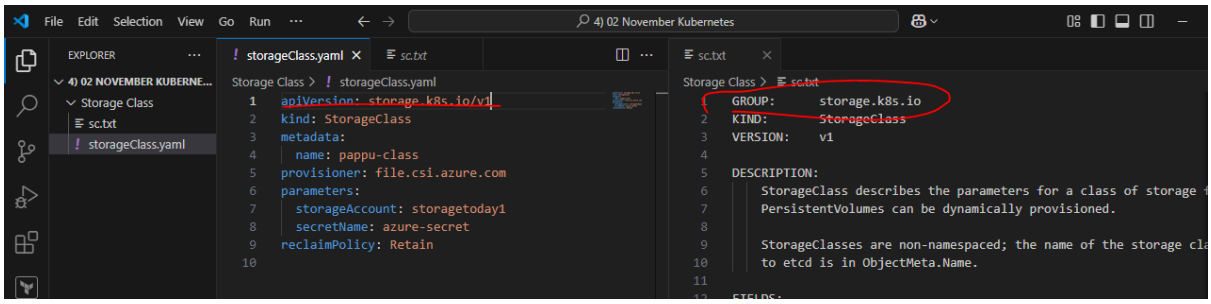
```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> az account set --subscription 48f88df7-0d53-4866-a66f-82eb0ac469e3
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> az aks get-credentials --resource-group rgtoday --name k8today --overwrite-existing
Merged "k8today" as current context in C:\Users\HP\.kube\config
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

27) Bring doc of storage class

Kubectl explain storageclasses --recursive > sc.txt

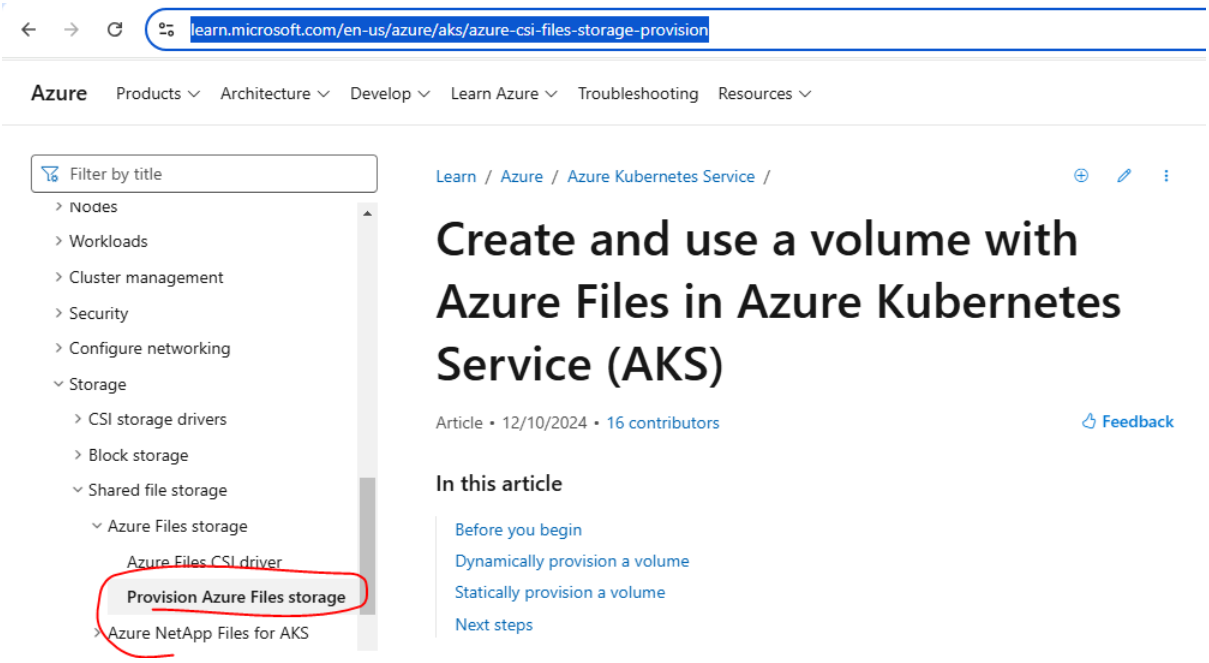


NOTE : 1) Whenever group comes in our doc then it will be apiVersion not v1



28) **provisioner:** = means kaha se provision krna hai, azure storage acc ke file share se, ya aws ki s3 bucket se ya gcp ki bucket se

29) SEARCH = https://learn.microsoft.com/en-us/azure/aks/azure-csi-files-storage-provision



```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: my-azurefile
provisioner: file.csi.azure.com # replace with "kubern
allowVolumeExpansion: true
mountOptions:
 - dir_mode=0777
```

30) Create storage account

Create secret also

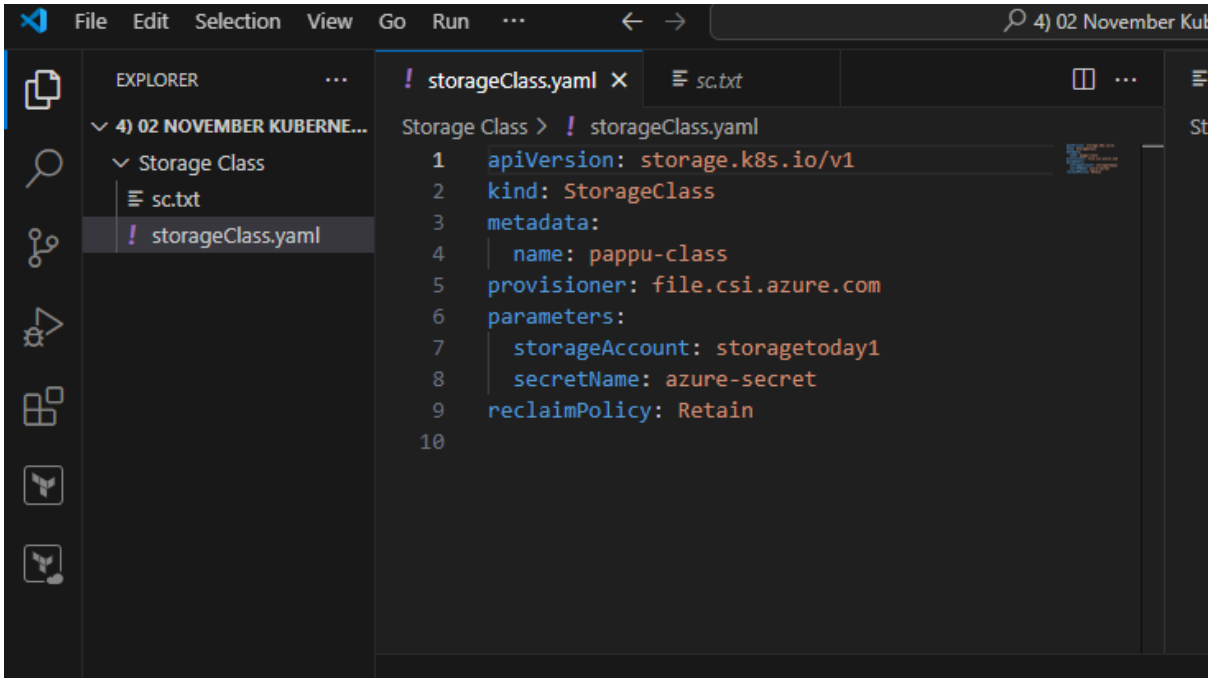
 # kubectl create secret generic azure-secret --from-literal=azurestorageaccountname=<...> --from-literal=azurestorageaccountkey=<...>

kubectl create secret generic azure-secret --from-literal=azurestorageaccountname=storagetoday1 --from-literal=azurestorageaccountkey=8B4Sq5RVc+pfihtKc+EKe2JBkYChHbRj0Z5KM9d+gR5NpIkFKVq43+KGM2gpHKksgd9IMRZ4iurT+ASt+HiJXw==

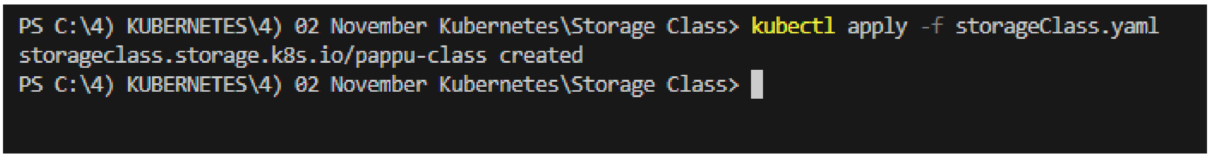
```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl create secret generic azure-secret --from-literal=azurestorageaccountname=storagetoday1 --
-from-literal=azurestorageaccountkey=8B4Sq5RVc+pfihtKc+EKe2JBkYChHbRj0Z5KM9d+gR5NpIkFKVq43+KGM2gpHKksgd9IMRZ4iurT+ASt+HiJXw==
secret/azure-secret created
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

31) allowVolumeExpansion: true = to increase volume in it

32) Now storageClass.yaml is ready to create pv, storage account automatically.

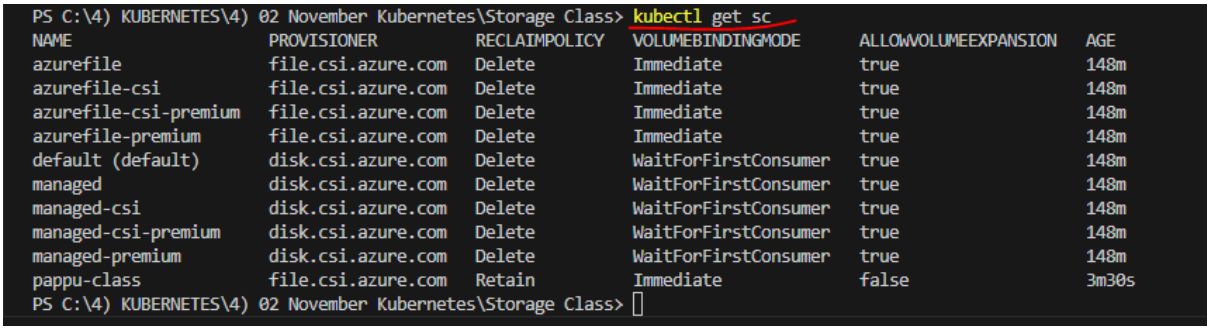


33) **kubectl apply -f storageClass.yaml** = Create storage class



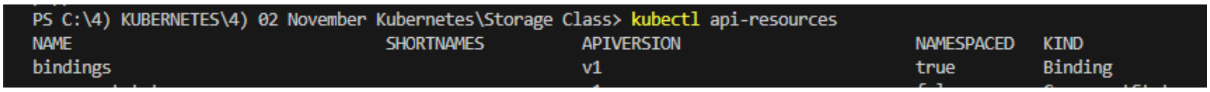
```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl apply -f storageClass.yaml
storageclass.storage.k8s.io/pappu-class created
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

34) **kubectl get sc**

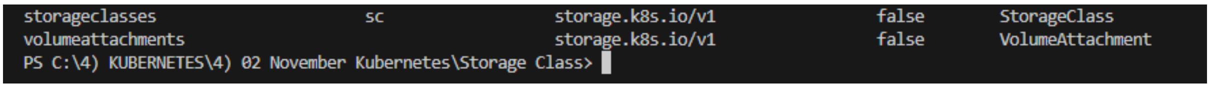


```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get sc
NAME                    PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
azurefile               file.csi.azure.com   Delete          Immediate              true                   148m
azurefile-csi           file.csi.azure.com   Delete          Immediate              true                   148m
azurefile-csi-premium   file.csi.azure.com   Delete          Immediate              true                   148m
azurefile-premium       file.csi.azure.com   Delete          Immediate              true                   148m
default (default)       disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   148m
managed                 disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   148m
managed-csi             disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   148m
managed-csi-premium     disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   148m
managed-premium         disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   148m
pappu-class             file.csi.azure.com   Retain          Immediate              false                  3m30s
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

35) Storage classes cannot be made in any specific name space its actually made for whole cluster. We can verify below seeing namespace colum for storageclasses is false



```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl api-resources
NAME                                SHORTNAMES          APIVERSION                          NAMESPACED   KIND
bindings                                                v1                                  true         Binding
```



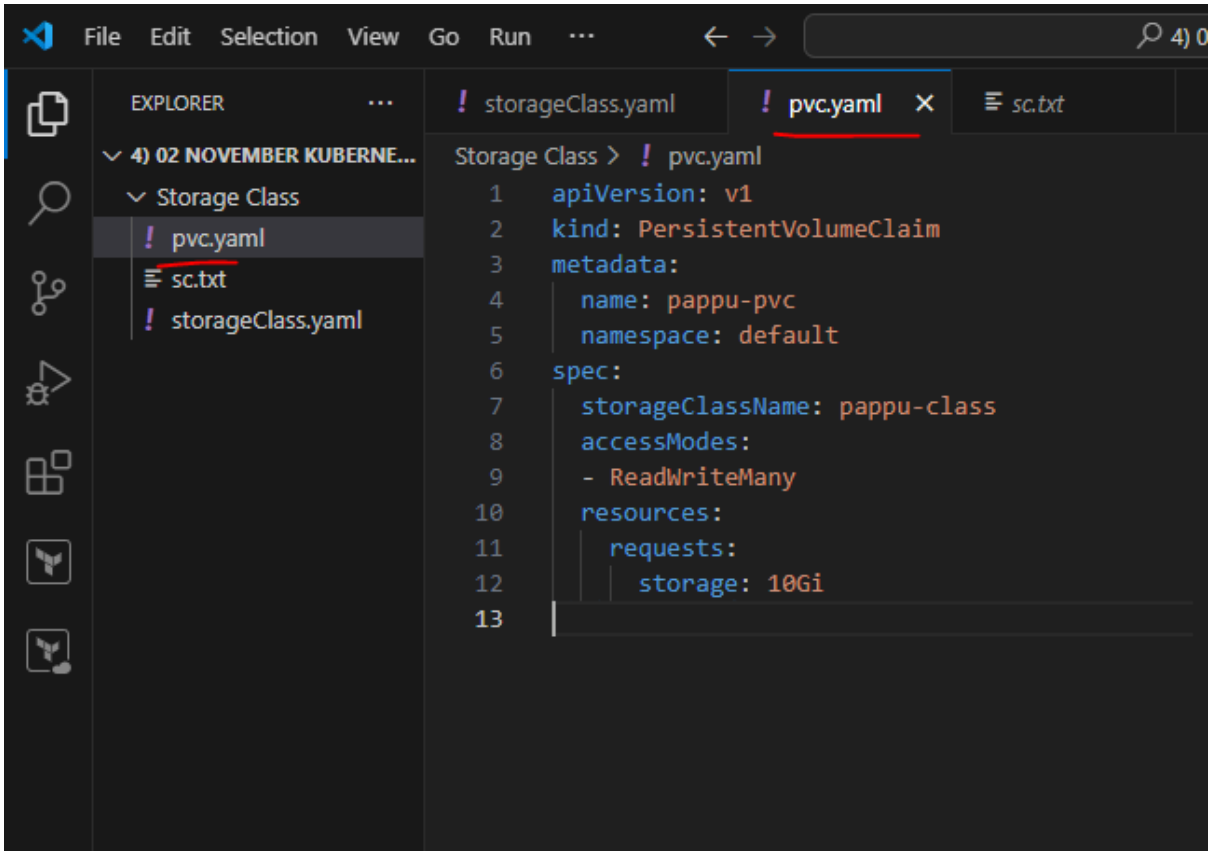
```
storageclasses                     sc                   storage.k8s.io/v1                   false        StorageClass
volumeattachments                                       storage.k8s.io/v1                   false        VolumeAttachment
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

++

AGENDA – CREATE PVC.YAML

NOTE: 1) PV are at cluster level and PVC are at namespace level

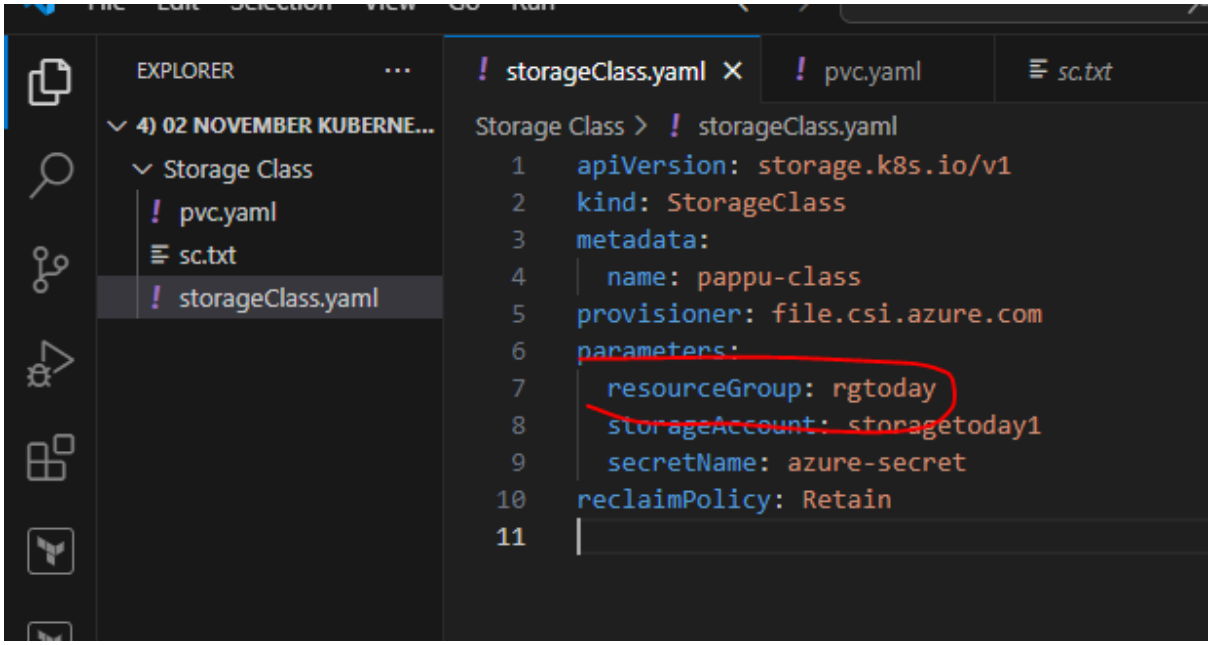
1) Create pvc.yaml file



2) **kubectl apply -f pvc.yaml =** creating pvc

```
No resources found
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl apply -f pvc.yaml
persistentvolumeclaim/pappu-pvc created
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

3) We had got error because we didn't mention rg



4) Deleting

kubectl get pvc

kubectl get sc

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get pvc
NAME        STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
pappu-pvc   Pending                                      pappu-class    <unset>                 9m11s
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get sc
NAME                    PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
azurefile               file.csi.azure.com   Delete          Immediate              true                   179m
azurefile-csi           file.csi.azure.com   Delete          Immediate              true                   179m
azurefile-csi-premium   file.csi.azure.com   Delete          Immediate              true                   179m
azurefile-premium       file.csi.azure.com   Delete          Immediate              true                   179m
default (default)       disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   179m
managed                 disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   179m
managed-csi             disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   179m
managed-csi-premium     disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   179m
managed-premium         disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   179m
pappu-class             file.csi.azure.com   Retain          Immediate              false                  34m
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

Kubectl delete pvc pappu-pvc

Kubectl delete sc pappu-class

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> Kubectl delete pvc pappu-pvc
persistentvolumeclaim "pappu-pvc" deleted
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> Kubectl delete sc pappu-class
storageclass.storage.k8s.io "pappu-class" deleted
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

5) Added rg now again create sc and pvc

6) Not able to do above agenda as assignment is create a user managed identity. Then attach it to cluster and then everything will be done for above agenda i.e. automatically create pv and file share in storage account.

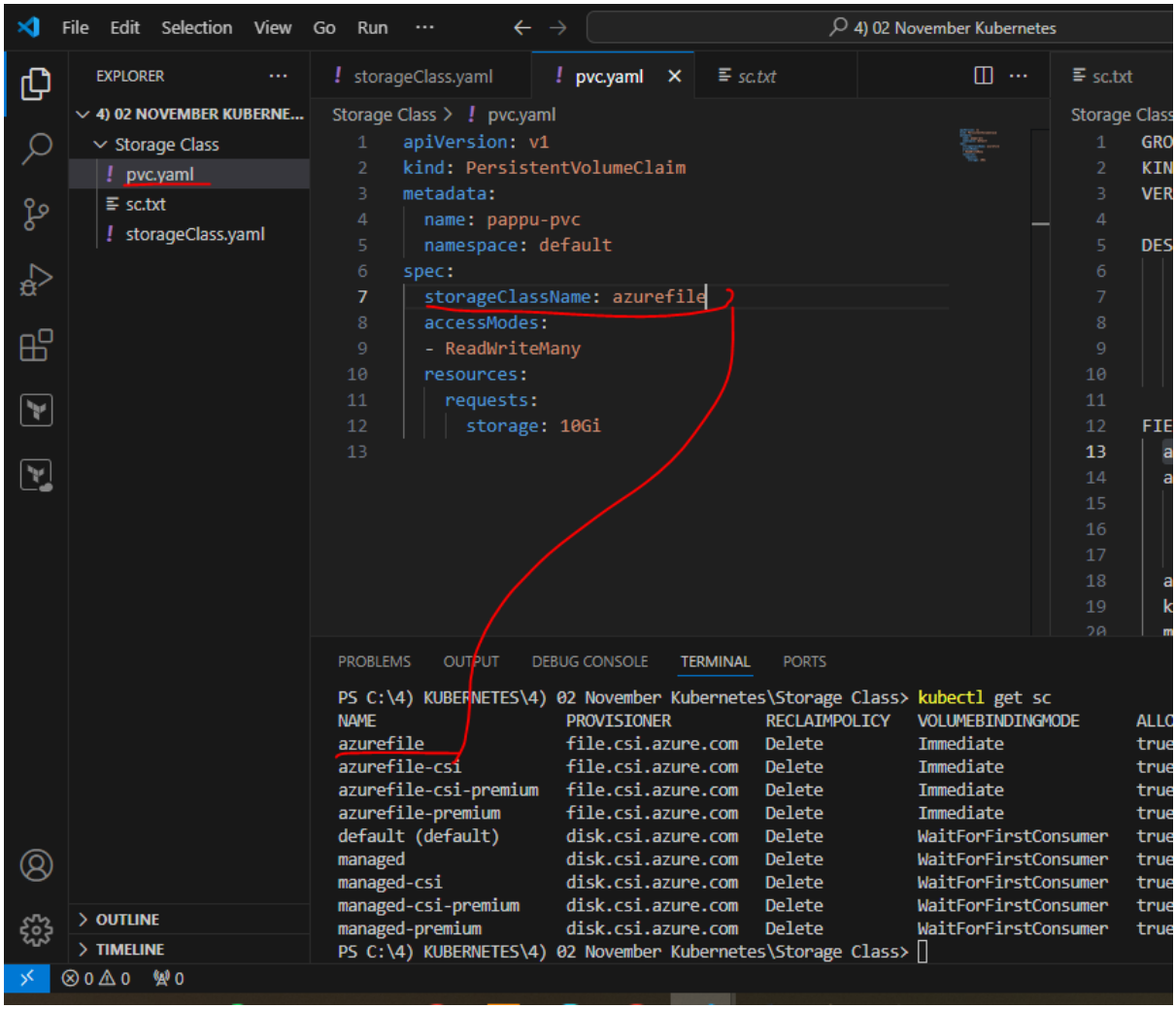
++

AGENDA – USING AUTOMATICALLY MADE STORAGE CLASS

1) **kubectl get sc** = already created sc

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get sc
NAME                    PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
azurefile               file.csi.azure.com   Delete          Immediate              true                   4h17m
azurefile-csi           file.csi.azure.com   Delete          Immediate              true                   4h17m
azurefile-csi-premium   file.csi.azure.com   Delete          Immediate              true                   4h17m
azurefile-premium       file.csi.azure.com   Delete          Immediate              true                   4h17m
default (default)       disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   4h17m
managed                 disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   4h17m
managed-csi             disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   4h17m
managed-csi-premium     disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   4h17m
managed-premium         disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   4h17m
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

2) Use already created storage class



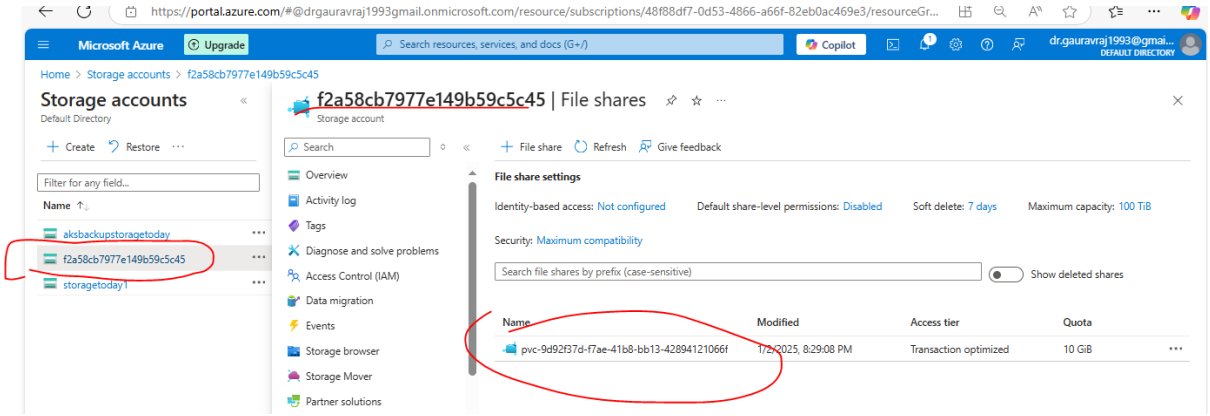
3) **kubectl apply -f pvc.yaml** = create pvc

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl apply -f pvc.yaml
persistentvolumeclaim/pappu-pvc created
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

4) **kubectl get pvc**

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get pvc
NAME        STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
pappu-pvc   Bound    pvc-9d92f37d-f7ae-41b8-bb13-42894121066f   10Gi       RWX            azurefile      <unset>                 37s
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

5) Now pvc got bounded and a storage account and file share is created automatically.

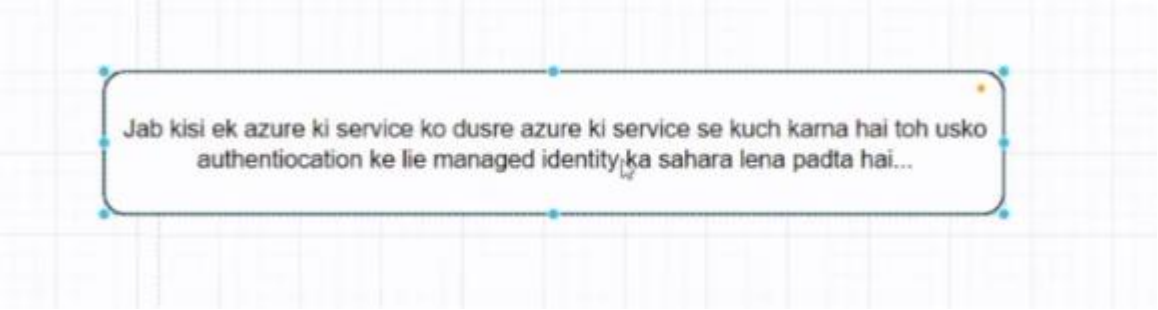


6)

++

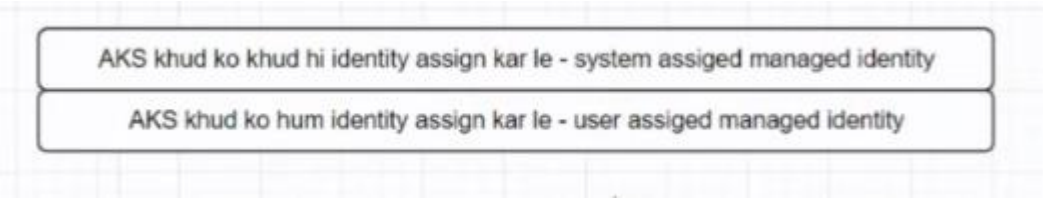
AGENDA – Assign managed identity to our aks cluster

1) Why managed identity is used?

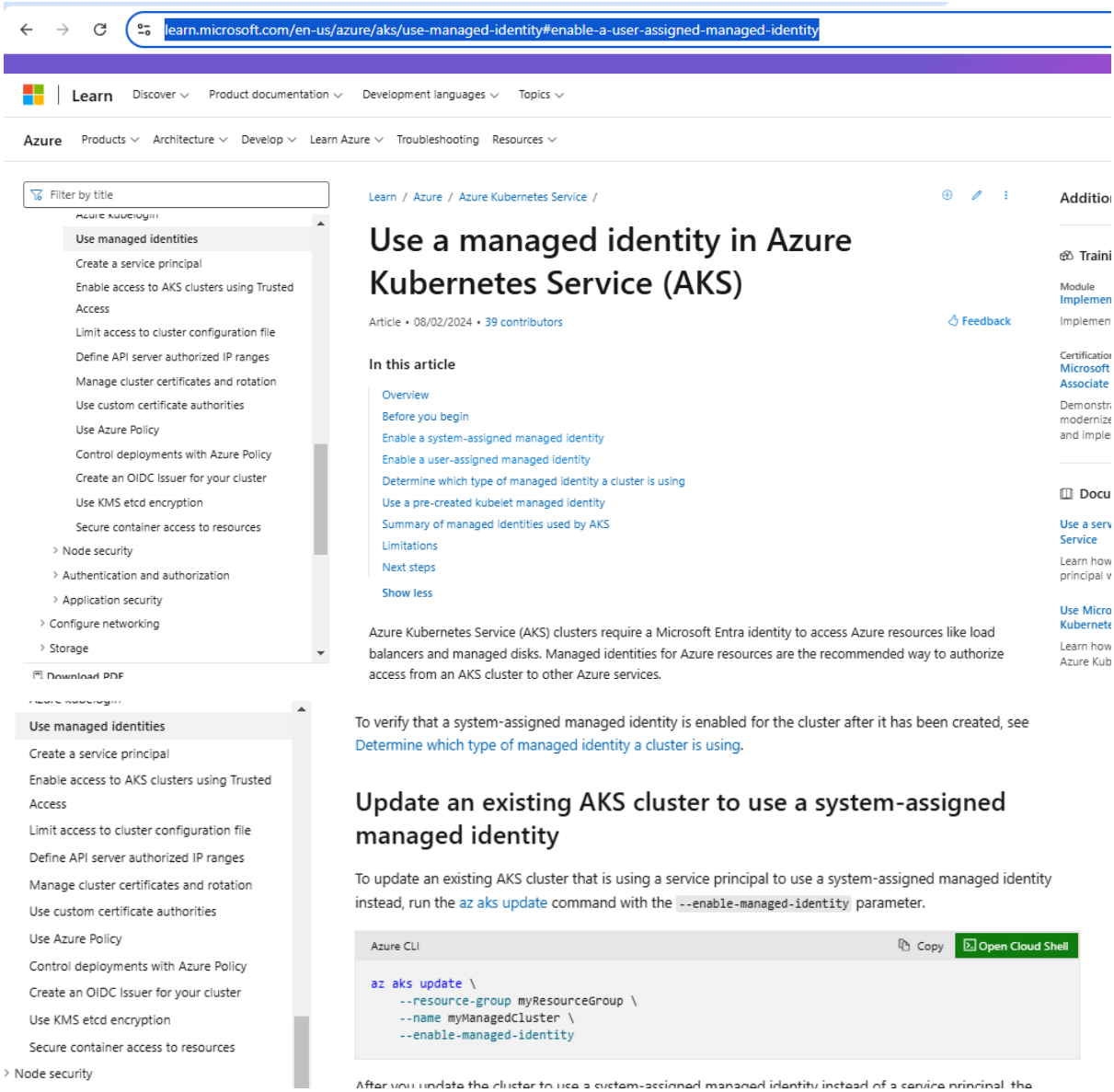


2) So aks ko azure ke storage account se baat krna hai to managed identity aks pr chipka denge

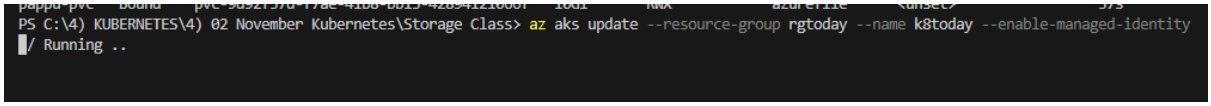
3) Managed identity are of two types – System assigned and user assigned



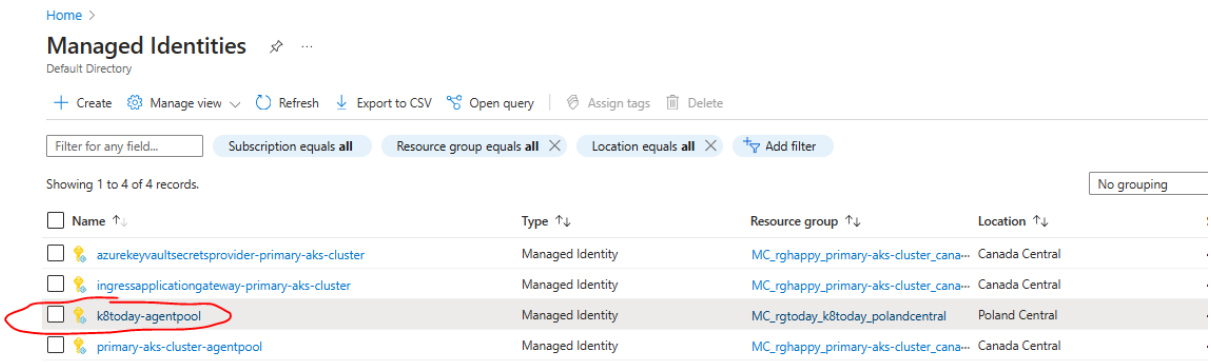
4) SEARCH = https://learn.microsoft.com/en-us/azure/aks/use-managed-identity#enable-a-user-assigned-managed-identity



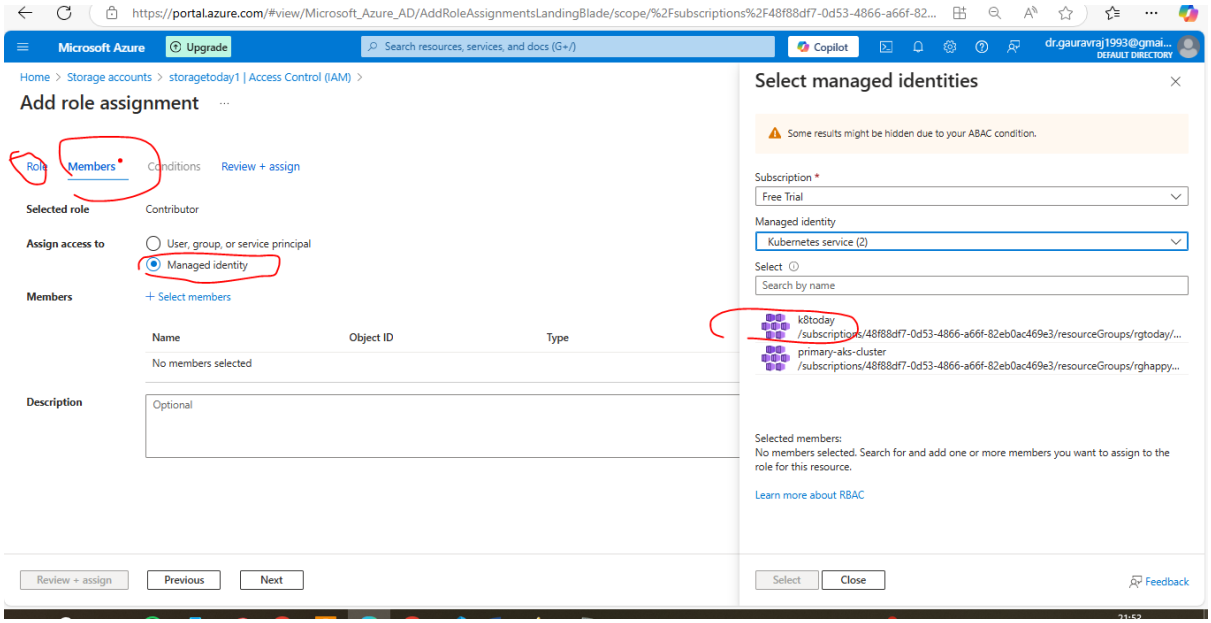
5) **az aks update --resource-group rgtoday --name k8today --enable-managed-identity** = run to Update an existing AKS cluster to use a system-assigned managed identity



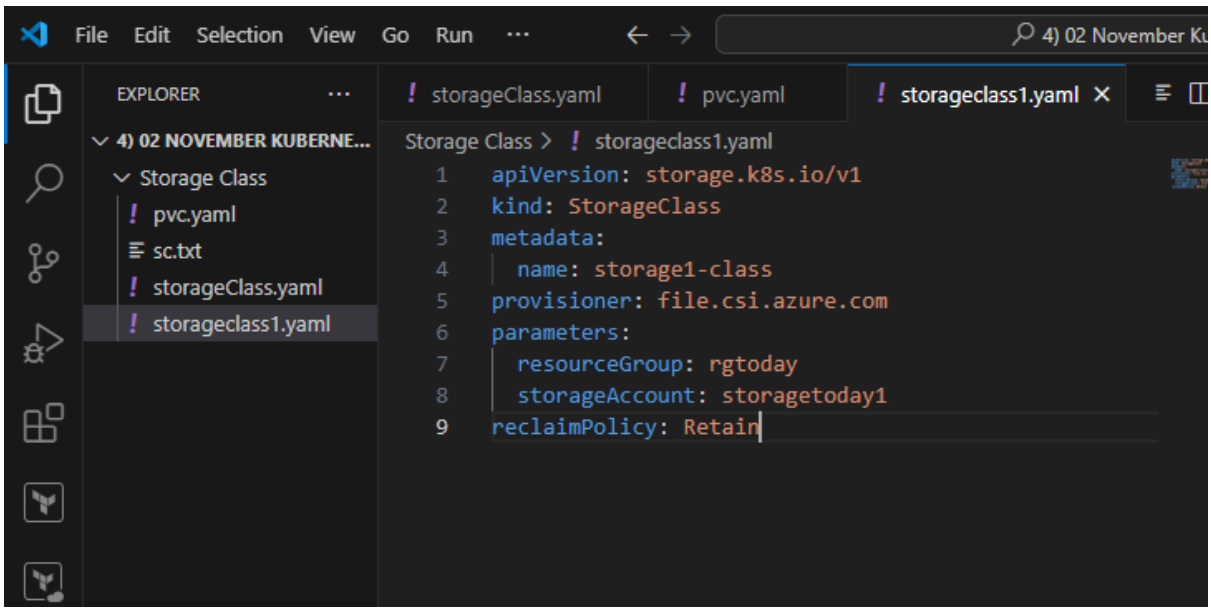
6) So now automatically managed identity created and it got attached to our k8s cluster



6) Go to storage account – acess control – give contributor role – select below details – review and assign



7) Create storageClass1.yaml



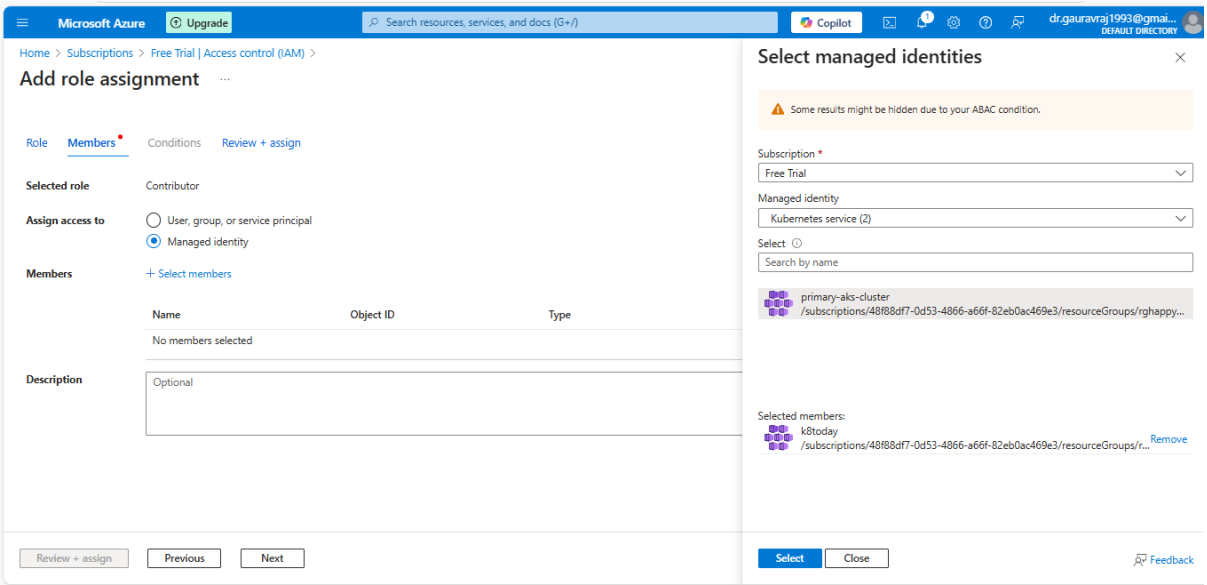
8) **kubectl apply -f storageclass1.yaml**

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl apply -f storageclass1.yaml
storageclass.storage.k8s.io/storage1-class created
```

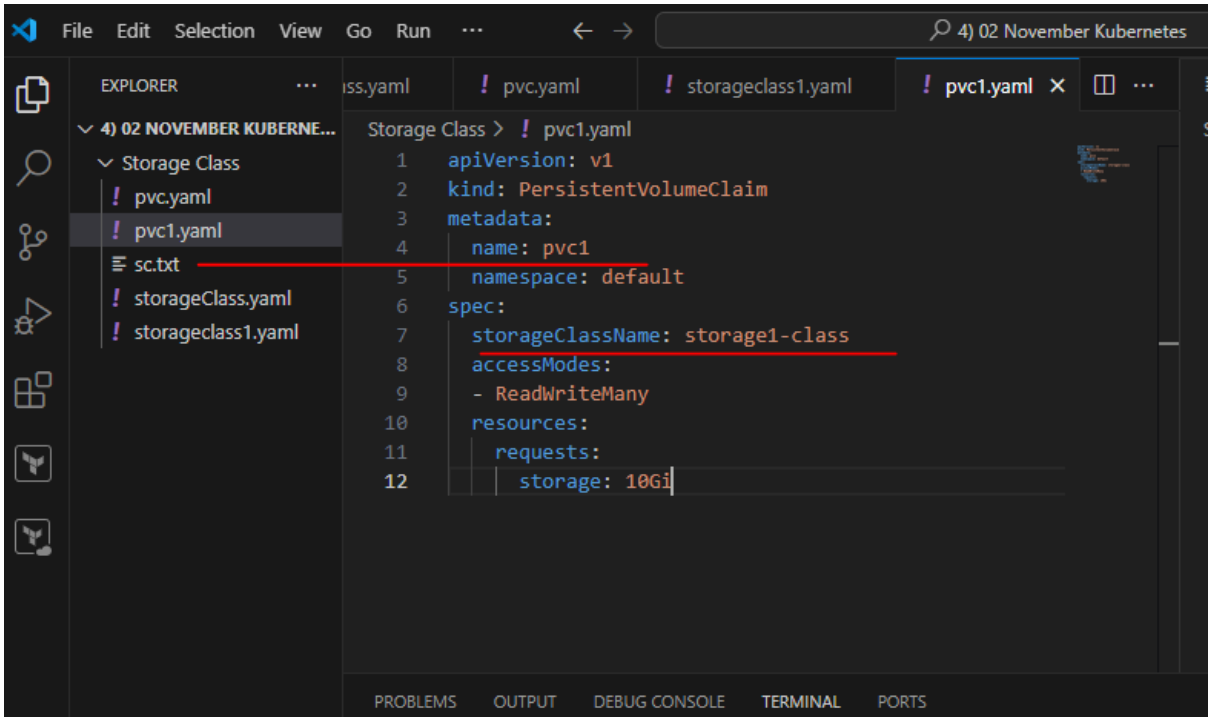
9) **kubectl get sc**

```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get sc
NAME                    PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
azurefile               file.csi.azure.com   Delete          Immediate              true                   5h55m
azurefile-csi           file.csi.azure.com   Delete          Immediate              true                   5h55m
azurefile-csi-premium   file.csi.azure.com   Delete          Immediate              true                   5h55m
azurefile-premium       file.csi.azure.com   Delete          Immediate              true                   5h55m
default (default)       disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   5h55m
managed                 disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   5h55m
managed-csi             disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   5h55m
managed-csi-premium     disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   5h55m
managed-premium         disk.csi.azure.com   Delete          WaitForFirstConsumer   true                   5h55m
storage1-class          file.csi.azure.com   Retain          Immediate              false                  49s
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class>
```

10) Go to subscription and give contributor access to aks cluster



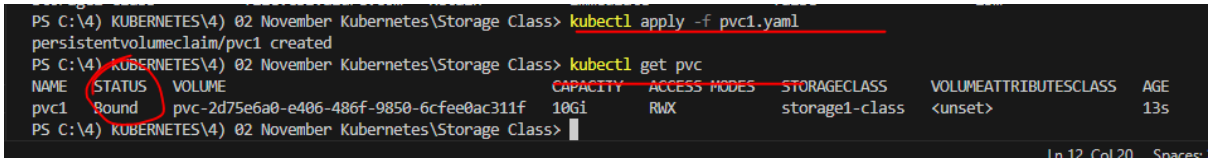
11) Create new pvc.yaml



```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc1
  namespace: default
spec:
  storageClassName: storage1-class
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

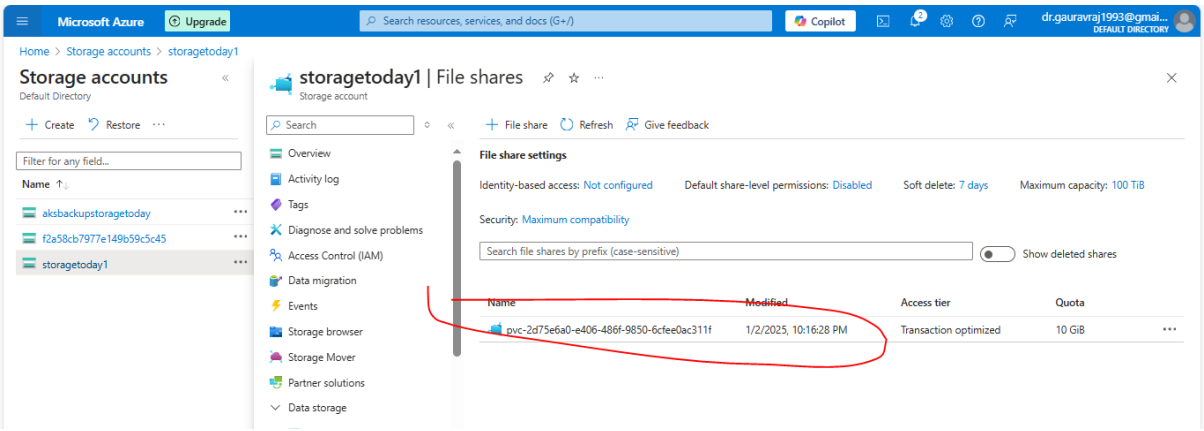
12) **kubectl apply -f pvc1.yaml**

kubectl get pvc



```
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl apply -f pvc1.yaml
persistentvolumeclaim/pvc1 created
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> kubectl get pvc
NAME   STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS     VOLUMEATTRIBUTESCLASS   AGE
pvc1   Bound    pvc-2d75e6a0-e406-486f-9850-6cfee0ac311f   10Gi       RWX            storage1-class   <unset>                 13s
PS C:\4) KUBERNETES\4) 02 November Kubernetes\Storage Class> 
```

This time its bounded means new file share created in storage account



Identity-based access: Not configured   Default share-level permissions: Disabled   Soft delete: 7 days   Maximum capacity: 100 TiB

Security: Maximum compatibility

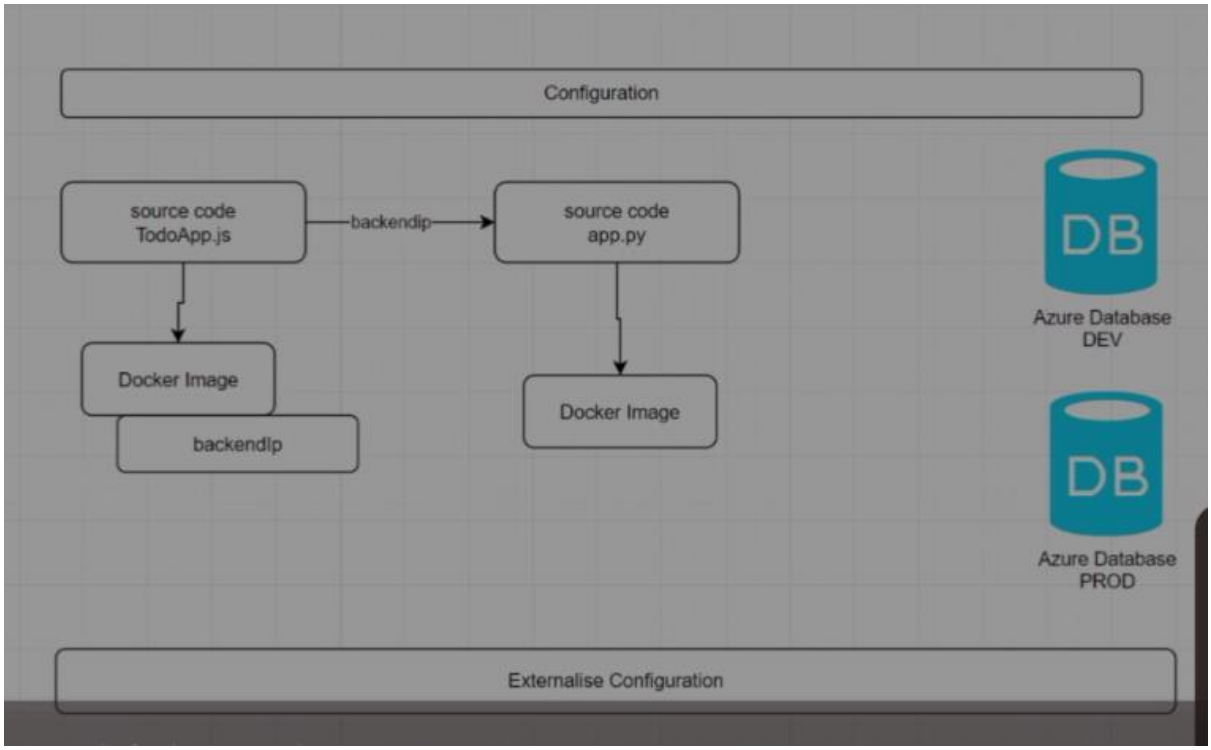
Name	Modified	Access tier	Quota
pvc-2d75e6a0-e406-486f-9850-6cfee0ac311f	1/2/2025, 10:16:28 PM	Transaction optimized	10 GiB

++

AGENDA – CONFIG MAPS

1) Config map is like space where we keep less sensitive things. For high sensitive things we use secrets.

2) 1) Configmap = It maps configurations which are not sensitive like that don't have password, key.


Configuration
source code
TodoApp.js
backendip
source code
app.py
Docker Image
backendIp
Docker Image
DB
Azure Database
DEV
DB
Azure Database
PROD
Externalise Configuration