

Kubernetes/AKS

1. Deploy Replica Set and Replication Controller, and deployment. Also learn the advantages and disadvantages of each.

```
ayush@ayush:~$ minikube start
😄 minikube v1.36.0 on Ubuntu 22.04
👉 Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.47 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
Verifying Kubernetes components...
🌟 Enabled addons:

❗ /usr/local/bin/kubectl is version 1.31.0, which may have incompatibilities with Kubernetes 1.33.1.
    ■ Want kubectl v1.33.1? Try 'minikube kubectl -- get pods -A'
🚀 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ayush@ayush:~$ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-5c84ff8d7f-8kpvx  1/1     Running   1 (7h ago)  7h59m
nginx-deployment-5c84ff8d7f-kvjpt  1/1     Running   1 (7h ago)  7h59m
nginx-deployment-5c84ff8d7f-tk6nw  1/1     Running   1 (7h ago)  7h59m
nginx-deployment-5c84ff8d7f-xlzj4  1/1     Running   0          7h34m
ayush@ayush:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE      VERSION
minikube  Ready    control-plane  5d7h    v1.33.1
ayush@ayush:~$ kubectl delete all --all
pod "nginx-deployment-5c84ff8d7f-8kpvx" deleted
pod "nginx-deployment-5c84ff8d7f-kvjpt" deleted
pod "nginx-deployment-5c84ff8d7f-tk6nw" deleted
pod "nginx-deployment-5c84ff8d7f-xlzj4" deleted
service "kubernetes" deleted
deployment.apps "nginx-deployment" deleted
```

```
ayush@ayush:~$ vim replication-controller.yaml
ayush@ayush:~$ cat replication-controller.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: my-rc
spec:
  replicas: 2
  selector:
    app: myapp
  template:
    metadata:
      labels:
        app: myapp
  spec:
    containers:
      - name: nginx
        image: nginx:1.21
      ports:
        - containerPort: 80
```

```

ayush@ayush:~$ kubectl apply -f replication-controller.yaml
replicationcontroller/my-rc created
ayush@ayush:~$ kubectl get rc
NAME    DESIRED   CURRENT   READY   AGE
my-rc   2          2          0       7s
ayush@ayush:~$ kubectl get rc -owide
NAME    DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES   SELECTOR
my-rc   2          2          0       16s   nginx        nginx:1.21   app=myapp
ayush@ayush:~$ kubectl get rc -w
NAME    DESIRED   CURRENT   READY   AGE
my-rc   2          2          0       25s
my-rc   2          2          1       27s
my-rc   2          2          2       32s
^Cayush@ayush:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-rc-f94t4  1/1    Running   0          44s
my-rc-gv6gs  1/1    Running   0          44s

```

```

ayush@ayush:~$ vim replica-set.yaml
ayush@ayush:~$ cat replica-set.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: my-rs
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: nginx
          image: nginx:1.21
          ports:
            - containerPort: 80

```

```

ayush@ayush:~$ kubectl apply -f replica-set.yaml
replicaset.apps/my-rs created
ayush@ayush:~$ kubectl get rs
NAME    DESIRED   CURRENT   READY   AGE
my-rs   3          3          3       10s
ayush@ayush:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
my-rc-f94t4  1/1    Running   0          5m45s
my-rc-gv6gs  1/1    Running   0          5m45s
my-rs-929g9  1/1    Running   0          20s
my-rs-df8dk  1/1    Running   0          20s
my-rs-nrrn9  1/1    Running   0          20s

```

```

ayush@ayush:~$ vim deployment.yaml
ayush@ayush:~$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: nginx
          image: nginx:1.21
          ports:
            - containerPort: 80
ayush@ayush:~$ kubectl apply -f deployment.yaml
deployment.apps/my-deployment created

```

```

ayush@ayush:~$ kubectl get deployments
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
my-deployment  3/3   3           3          11s
ayush@ayush:~$ kubectl get rs
NAME      DESIRED  CURRENT  READY  AGE
my-deployment-85cb5ff4c  3     3       3     4m5s
my-rs      3     3       3     12m
ayush@ayush:~$ kubectl get pods -owide
NAME      READY  STATUS  RESTARTS  AGE      IP      NODE  NOMINATED NODE  READINESS GATES
my-deployment-85cb5ff4c-8vz5g  1/1   Running  0  4m21s  10.244.0.75  minikube  <none>  <none>
my-deployment-85cb5ff4c-cdcz2  1/1   Running  0  4m21s  10.244.0.77  minikube  <none>  <none>
my-deployment-85cb5ff4c-qdhrn  1/1   Running  0  4m21s  10.244.0.76  minikube  <none>  <none>
my-rc-f94t4                    1/1   Running  0  17m    10.244.0.71  minikube  <none>  <none>
my-rc-gv6gs                     1/1   Running  0  17m    10.244.0.70  minikube  <none>  <none>
my-rs-929g9                     1/1   Running  0  12m    10.244.0.72  minikube  <none>  <none>
my-rs-df8dk                     1/1   Running  0  12m    10.244.0.73  minikube  <none>  <none>
my-rs-nrrn9                     1/1   Running  0  12m    10.244.0.74  minikube  <none>  <none>

```

```

ayush@ayush:~$ kubectl set image deployment/my-deployment nginx=nginx:1.22
deployment.apps/my-deployment image updated
ayush@ayush:~$ kubectl get pods -owide
NAME      READY  STATUS  RESTARTS  AGE      IP      NODE  NOMINATED NODE  READINESS GATES
my-deployment-57b897f796-6fmcj  0/1   ContainerCreating  0  6s    <none>  minikube  <none>  <none>
my-deployment-57b897f796-h74j5  0/1   ContainerCreating  0  6s    <none>  minikube  <none>  <none>
my-deployment-85cb5ff4c-cdcz2  1/1   Running  0  5m26s  10.244.0.77  minikube  <none>  <none>
my-deployment-85cb5ff4c-qdhrn  1/1   Running  0  5m26s  10.244.0.76  minikube  <none>  <none>
my-rc-f94t4                    1/1   Running  0  18m    10.244.0.71  minikube  <none>  <none>
my-rc-gv6gs                     1/1   Running  0  18m    10.244.0.70  minikube  <none>  <none>
my-rs-929g9                     1/1   Running  0  13m    10.244.0.72  minikube  <none>  <none>
my-rs-df8dk                     1/1   Running  0  13m    10.244.0.73  minikube  <none>  <none>
my-rs-nrrn9                     1/1   Running  0  13m    10.244.0.74  minikube  <none>  <none>
ayush@ayush:~$ kubectl rollout undo deployment/my-deployment
deployment.apps/my-deployment rolled back
ayush@ayush:~$ kubectl get pods -owide
NAME      READY  STATUS  RESTARTS  AGE      IP      NODE  NOMINATED NODE  READINESS GATES
my-deployment-85cb5ff4c-7kxsm  1/1   Running  0  6s    10.244.0.81  minikube  <none>  <none>
my-deployment-85cb5ff4c-fgm2j  1/1   Running  0  6s    10.244.0.82  minikube  <none>  <none>
my-deployment-85cb5ff4c-qdhrn  1/1   Running  0  6m1s  10.244.0.76  minikube  <none>  <none>
my-rc-f94t4                    1/1   Running  0  19m    10.244.0.71  minikube  <none>  <none>
my-rc-gv6gs                     1/1   Running  0  19m    10.244.0.70  minikube  <none>  <none>
my-rs-929g9                     1/1   Running  0  13m    10.244.0.72  minikube  <none>  <none>
my-rs-df8dk                     1/1   Running  0  13m    10.244.0.73  minikube  <none>  <none>
my-rs-nrrn9                     1/1   Running  0  13m    10.244.0.74  minikube  <none>  <none>
ayush@ayush:~$ 

```

```

ayush@ayush:~$ kubectl delete -f replication-controller.yaml
replicationcontroller "my-rc" deleted
ayush@ayush:~$ kubectl delete -f replica-set.yaml
replicaset.apps "my-rs" deleted
ayush@ayush:~$ kubectl delete -f deployment.yaml
deployment.apps "my-deployment" deleted
ayush@ayush:~$ kubectl get pods -owide
No resources found in default namespace.

```

2. Kubernetes service types (ClusterIP, NodePort, LoadBalancer)

```

ayush@ayush:~$ vim nginx-pod.yaml
ayush@ayush:~$ cat nginx-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  containers:
    - name: nginx
      image: nginx:1.21
      ports:
        - containerPort: 80
ayush@ayush:~$ kubectl apply -f nginx-pod.yaml
pod/nginx created
ayush@ayush:~$ kubectl get pods -owide
NAME    READY    STATUS    RESTARTS   AGE     IP           NODE    NOMINATED NODE   READINESS GATES
nginx  1/1     Running   0          14s    10.244.0.83  minikube <none>        <none>

```

```

ayush@ayush:~$ vim clusterip.yaml
ayush@ayush:~$ cat clusterip.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-clusterip
spec:
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
    type: ClusterIP
ayush@ayush:~$ kubectl apply -f clusterip.yaml
service/nginx-clusterip created
ayush@ayush:~$ kubectl get pods -owide
NAME    READY    STATUS    RESTARTS   AGE     IP           NODE    NOMINATED NODE   READINESS GATES
nginx  1/1     Running   0          2m21s  10.244.0.83  minikube <none>        <none>
ayush@ayush:~$ kubectl get services
NAME            TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
kubernetes      ClusterIP  10.96.0.1    <none>       443/TCP   15h
nginx-clusterip ClusterIP  10.101.89.70  <none>       80/TCP    26s

```

```

ayush@ayush:~$ kubectl run curlpod --image=curlimages/curl -i --restart=Never -- sh
If you don't see a command prompt, try pressing enter.
~ $ curl http://nginx-clusterip
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
~ $ ^C

~ $ exit
pod default/curlpod terminated (Error)

```

```

ayush@ayush:~$ vim nodeport.yaml
ayush@ayush:~$ cat nodeport.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 31000
    type: NodePort
ayush@ayush:~$ kubectl apply -f nodeport.yaml
service/nginx-nodeport created
ayush@ayush:~$ kubectl get nodes -owide
NAME     STATUS   ROLES    AGE     VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE          KERNEL-VERSION   CONTAINER-RUNTIME
minikube Ready    control-plane   5d22h   v1.33.1   192.168.49.2   <none>        Ubuntu 22.04.5 LTS   6.8.0-60-generic   docker://28.1.1

```

```

ayush@ayush:~$ curl http://192.168.49.2:31000
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```



```

ayush@ayush:~$ vim loadbalancer.yaml
ayush@ayush:~$ cat loadbalancer.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-loadbalancer
spec:
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
    type: LoadBalancer
ayush@ayush:~$ kubectl apply -f loadbalancer.yaml
service/nginx-loadbalancer created
ayush@ayush:~$ kubectl get svc nginx-loadbalancer
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
nginx-loadbalancer  LoadBalancer  10.108.124.68 <pending>   80:30306/TCP  33s
ayush@ayush:~$ kubectl get svc nginx-loadbalancer
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
nginx-loadbalancer  LoadBalancer  10.108.124.68 <pending>   80:30306/TCP  80s

```

```

ayush@ayush:~$ minikube tunnel
[sudo] password for ayush:
Status:
  machine: minikube
  pid: 66881
  route: 10.96.0.0/12 -> 192.168.49.2
  minikube: Running
  services: [nginx-loadbalancer]

errors:
  minikube: no errors
  router: no errors
  loadbalancer emulator: no errors

```

```

ayush@ayush: $ kubectl get svc nginx-loadbalancer
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
nginx-loadbalancer   LoadBalancer   10.108.124.68  10.108.124.68  80:30306/TCP  3m28s
ayush@ayush: $

```



3. PersistentVolume (PV) and PersistentVolumeClaim (PVC)

```

ayush@ayush:~$ vim pv.yaml
ayush@ayush:~$ cat pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /mnt/data
ayush@ayush:~$ kubectl apply -f pv.yaml
persistentvolume/my-pv created
ayush@ayush:~$ kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS     CLAIM   STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
my-pv    1Gi        RWO            Retain       Available   <unset>   <unset>          13s
ayush@ayush:~$ vim pvc.yaml
ayush@ayush:~$ cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

```
ayush@ayush:~$ kubectl apply -f pvc.yaml
persistentvolumeclaim/my-pvc created
ayush@ayush:~$ kubectl get pvc
NAME      STATUS  VOLUME           CAPACITY  ACCESS MODES  STORAGECLASS  VOLUME ATTRIBUTESCLASS  AGE
my-pvc   Bound   pvc-0473aee1-5e26-4d4a-b802-0e70a7854d0b  1Gi        RWO          standard      <unset>          12s
```

```
ayush@ayush:~$ vim pvcpod.yaml
ayush@ayush:~$ cat pvcpod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-pvc
spec:
  containers:
    - name: app
      image: nginx
      volumeMounts:
        - mountPath: /usr/share/nginx/html
          name: my-storage
  volumes:
    - name: my-storage
      persistentVolumeClaim:
        claimName: my-pvc
ayush@ayush:~$ kubectl apply -f pvcpod.yaml
pod/pod-pvc created
```

```
ayush@ayush:~$ kubectl get pods
NAME      READY  STATUS    RESTARTS  AGE
nginx    1/1    Running   0          51m
pod-pvc  1/1    Running   0          57s
ayush@ayush:~$ kubectl exec -it pod-pvc -- /bin/bash
root@pod-pvc:/# echo "hi from inside of pvc" > /usr/share/nginx/html/index.html
root@pod-pvc:/# kubectl port-forward pod/pod-pvc 8080:80
bash: kubectl: command not found
root@pod-pvc:/# exit
exit
command terminated with exit code 127
ayush@ayush:~$ kubectl port-forward pod/pod-pvc 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```

```
ayush@ayush:~$ curl http://localhost:8080
hi from inside of pvc
ayush@ayush:~$
```



4. Managing Kubernetes with Azure Kubernetes Service (AKS), Creating and managing AKS clusters, Scaling and upgrading AKS clusters

```
ayush [ ~ ]$ az group create --name resource-aks --location eastus
{
  "id": "/subscriptions/4af928af-144b-44ea-a5ad-9fe8d622bd21/resourceGroups/resource-aks",
  "location": "eastus",
  "managedBy": null,
  "name": "resource-aks",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

```
ayush [ ~ ]$ az aks create --resource-group resource-aks --name akscluster --node-count 3 --enable-cluster-autoscaler --min-count 1 --max-count 5 --node-vm-size Standard_B2s --generate-ssh-key
{
  "aadProfile": null,
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "capacityReservationGroupId": null,
      "count": 3,
      "creationData": null,
      "currentOrchestratorVersion": "1.31.9",
      "eTag": null,
      "enableAutoScaling": true,
      "enableEncryptionAtHost": false,
      "enableEips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": false,
      "gpuInstanceProfile": null,
      "gpuProfile": null,
      "hostGroupId": null,
      "kubeletConfig": null,
      "kubeletDiskType": "OS",
      "linuxOsConfig": null,
      "maxCount": 5,
      "maxPods": 250,
      "messageOfTheDay": null,
      "minCount": 1,
      "mode": "System",
      "name": "nodepool1",
      "networkProfile": null,
      "nodeImageVersion": "AKSUbuntu-2204gen2containerd-202506.03.0",
      "nodeLabels": null,
      "nodePublicIpPrefixId": null
    }
  ],
  "dnsPrefix": "akscluster",
  "nodeResourceGroup": "resource-aks"
}
```

```

        "workloadIdentity": null
    },
    "serviceMeshProfile": null,
    "servicePrincipalProfile": {
        "clientId": "msi",
        "secret": null
    },
    "sku": {
        "name": "Base",
        "tier": "Free"
    },
    "storageProfile": {
        "blobCsiDriver": null,
        "diskCsiDriver": {
            "enabled": true
        },
        "fileCsiDriver": {
            "enabled": true
        },
        "snapshotController": {
            "enabled": true
        }
    },
    "supportPlan": "KubernetesOfficial",
    "systemData": null,
    "tags": null,
    "type": "Microsoft.ContainerService/ManagedClusters",
    "upgradeSettings": null,
    "windowsProfile": null,
    "workloadAutoScalerProfile": {
        "keda": null,
        "verticalPodAutoscaler": null
    }
}

```

```

ayush [ ~ ]$ az aks get-credentials --resource-group resource-aks --name akscluster
Merged "akscluster" as current context in /home/ayush/.kube/config
ayush [ ~ ]$ kubectl get nodes
NAME                  STATUS  ROLES   AGE     VERSION
aks-nodepool1-18391766-vmss000000  Ready   <none>  3m11s  v1.31.9
aks-nodepool1-18391766-vmss000001  Ready   <none>  3m6s   v1.31.9
aks-nodepool1-18391766-vmss000002  Ready   <none>  2m58s  v1.31.9
ayush [ ~ ]$ ls
ayush [ ~ ]$ vim nginx-deployment.yaml
ayush [ ~ ]$ cat nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
  spec:
    containers:
      - name: nginx
        image: nginx:1.21
        ports:
          - containerPort: 80

```

```

ayush [ ~ ]$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deploy created
ayush [ ~ ]$ kubectl get pods -owide
NAME                      READY   STATUS    RESTARTS   AGE     IP           NODE      NOMINATED NODE   READINESS GATES
nginx-deploy-7cfcf9b64b-qqdkr  1/1    Running   0          11s    10.244.2.218  aks-nodepool1-18391766-vms000002  <none>        <none>
nginx-deploy-7cfcf9b64b-xj86t  1/1    Running   0          11s    10.244.1.199  aks-nodepool1-18391766-vms000001  <none>        <none>
ayush [ ~ ]$ kubectl expose deployment nginx-deploy --port 80 --target-port 80 --type LoadBalancer
service/nginx-deploy exposed
ayush [ ~ ]$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.0.0.1      <none>        443/TCP      17m
nginx-deploy  LoadBalancer  10.0.185.242  <pending>    80:30586/TCP  8s

```

```

ayush [ ~ ]$ az aks update --resource-group resource-aks --name akscluster --disable-cluster-autoscaler
{
  "aadProfile": null,
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "capacityReservationGroupId": null,
      "count": 2,
      "creationData": null,
      "currentOrchestratorVersion": "1.31.9",
      "eTag": null,
      "enableAutoScaling": false,
      "enableEncryptionAtHost": false,
      "enableFips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": false,
      "gpuInstanceProfile": null,
      "gpuProfile": null,
      "hostGroupId": null,
      "kubeletConfig": null,
      "kubeletDiskType": "OS",
      "linuxOsConfig": null,
      "maxCount": null,
      "maxPods": 250,
      "messageOfTheDay": null,
      "minCount": null,
      "mode": "System",
      "name": "nodepool1",
      "networkProfile": null,
      "nodeImageVersion": "AKSUBuntu-2204gen2containerd-202506.03.0",
      "nodeLabels": null,
      "nodePublicIpPrefixId": null,
      "nodeTaints": null
    }
  ]
}

```

```
        "workloadIdentity": null
    },
    "serviceMeshProfile": null,
    "servicePrincipalProfile": {
        "clientId": "msi",
        "secret": null
    },
    "sku": {
        "name": "Base",
        "tier": "Free"
    },
    "storageProfile": {
        "blobCsiDriver": null,
        "diskCsiDriver": {
            "enabled": true
        },
        "fileCsiDriver": {
            "enabled": true
        },
        "snapshotController": {
            "enabled": true
        }
    },
    "supportPlan": "KubernetesOfficial",
    "systemData": null,
    "tags": null,
    "type": "Microsoft.ContainerService/ManagedClusters",
    "upgradeSettings": null,
    "windowsProfile": null,
    "workloadAutoScalerProfile": {
        "keda": null,
        "verticalPodAutoscaler": null
    }
}
```

```
ayush [ ~ ]$ az aks scale --resource-group resource-aks --name akscluster --node-count 5
{
    "aadProfile": null,
    "addonProfiles": null,
    "agentPoolProfiles": [
        {
            "availabilityZones": null,
            "capacityReservationGroupId": null,
            "count": 5,
            "creationData": null,
            "currentOrchestratorVersion": "1.31.9",
            "eTag": null,
            "enableAutoScaling": false,
            "enableEncryptionAtHost": false,
            "enableFips": false,
            "enableNodePublicIp": false,
            "enableUltraSsd": false,
            "gpuInstanceProfile": null,
            "gpuProfile": null,
            "hostGroupId": null,
            "kubeletConfig": null,
            "kubeletDiskType": "OS",
            "linuxOsConfig": null,
            "maxCount": null,
            "maxPods": 250,
            "messageOfTheDay": null,
            "minCount": null,
            "mode": "System",
            "name": "nodepool1",
            "networkProfile": null,
            "nodeImageVersion": "AKSUbuntu-2204gen2containerd-202506.03.0",
            "nodeLabels": null,
            "nodePublicIpPrefixId": null,
            "nodeTaints": null
        }
    ]
}
```

```
        "workloadIdentity": null
    },
    "serviceMeshProfile": null,
    "servicePrincipalProfile": {
        "clientId": "msi",
        "secret": null
    },
    "sku": {
        "name": "Base",
        "tier": "Free"
    },
    "storageProfile": {
        "blobCsiDriver": null,
        "diskCsiDriver": {
            "enabled": true
        },
        "fileCsiDriver": {
            "enabled": true
        },
        "snapshotController": {
            "enabled": true
        }
    },
    "supportPlan": "KubernetesOfficial",
    "systemData": null,
    "tags": null,
    "type": "Microsoft.ContainerService/ManagedClusters",
    "upgradeSettings": null,
    "windowsProfile": null,
    "workloadAutoScalerProfile": {
        "keda": null,
        "verticalPodAutoscaler": null
    }
}
```

```
ayush [ ~ ]$ az aks show --resource-group resource-aks --name akscluster
{
    "aadProfile": null,
    "addonProfiles": null,
    "agentPoolProfiles": [
        {
            "availabilityZones": null,
            "capacityReservationGroupId": null,
            "count": 5,
            "creationData": null,
            "currentOrchestratorVersion": "1.31.9",
            "eTag": null,
            "enableAutoScaling": false,
            "enableEncryptionAtHost": false,
            "enableFips": false,
            "enableNodePublicIp": false,
            "enableUltraSsd": false,
            "gpuInstanceProfile": null,
            "gpuProfile": null,
            "hostGroupId": null,
            "kubeletConfig": null,
            "kubeletDiskType": "OS",
            "linuxOsConfig": null,
            "maxCount": null,
            "maxPods": 250,
            "messageOfTheDay": null,
            "minCount": null,
            "mode": "System",
            "name": "nodepool1",
            "networkProfile": null,
            "nodeImageVersion": "AKSUbuntu-2204gen2containerd-202506.03.0",
            "nodeLabels": null,
            "nodePublicIpPrefixId": null,
            "nodeTaints": null,
            "osDiskSizeGb": 30,
            "osType": "Linux",
            "podCidr": "10.10.0.0/16",
            "pullSecret": null,
            "resourceRequirements": null,
            "role": "Agent",
            "vmSize": "Standard_DS2_v2"
        }
    ],
    "apiserverIP": "10.0.0.11",
    "apiserverPort": 44667,
    "controlPlaneRole": "Control Plane",
    "controlPlaneVmSize": "Standard_DS2_v2",
    "dnsName": "akscluster.ayush.local",
    "dnsZone": "ayush.local",
    "fqdn": "akscluster.ayush.local",
    "fullyQualifiedDomainName": "akscluster.ayush.local",
    "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster",
    "location": "West US 2",
    "managedClusterProfile": {
        "apiServerIp": "10.0.0.11",
        "apiServerPort": 44667,
        "controlPlaneVmSize": "Standard_DS2_v2",
        "dnsName": "akscluster.ayush.local",
        "dnsZone": "ayush.local",
        "fqdn": "akscluster.ayush.local",
        "fullyQualifiedDomainName": "akscluster.ayush.local",
        "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster/managedClusterProfile",
        "location": "West US 2",
        "managedClusterType": "User Managed",
        "nodePools": [
            {
                "count": 5,
                "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster/nodePools/nodepool1",
                "label": "nodepool1",
                "location": "West US 2",
                "name": "nodepool1",
                "nodeCount": 5,
                "nodeVmSize": "Standard_DS2_v2",
                "osType": "Linux",
                "type": "VirtualMachineScaleSet"
            }
        ],
        "nodePools": [
            {
                "count": 5,
                "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster/nodePools/nodepool1",
                "label": "nodepool1",
                "location": "West US 2",
                "name": "nodepool1",
                "nodeCount": 5,
                "nodeVmSize": "Standard_DS2_v2",
                "osType": "Linux",
                "type": "VirtualMachineScaleSet"
            }
        ],
        "type": "VirtualMachineScaleSet"
    },
    "nodePools": [
        {
            "count": 5,
            "id": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster/nodePools/nodepool1",
            "label": "nodepool1",
            "location": "West US 2",
            "name": "nodepool1",
            "nodeCount": 5,
            "nodeVmSize": "Standard_DS2_v2",
            "osType": "Linux",
            "type": "VirtualMachineScaleSet"
        }
    ],
    "nodeVmSize": "Standard_DS2_v2",
    "osType": "Linux",
    "resourceGroup": "resource-aks",
    "type": "Microsoft.ContainerService/managedClusters"
}
```

```
        "defender": null,
        "imageCleaner": null,
        "workloadIdentity": null
    },
    "serviceMeshProfile": null,
    "servicePrincipalProfile": {
        "clientId": "msi"
    },
    "sku": {
        "name": "Base",
        "tier": "Free"
    },
    "storageProfile": {
        "blobCsiDriver": null,
        "diskCsiDriver": {
            "enabled": true
        },
        "fileCsiDriver": {
            "enabled": true
        },
        "snapshotController": {
            "enabled": true
        }
    },
    "supportPlan": "KubernetesOfficial",
    "systemData": null,
    "type": "Microsoft.ContainerService/ManagedClusters",
    "upgradeSettings": null,
    "windowsProfile": null,
    "workloadAutoScalerProfile": {
        "keda": null,
        "verticalPodAutoscaler": null
    }
}
```

```

ayush [ ~ ]$ az aks get-upgrades --resource-group resource-aks --name akscluster
{
  "agentPoolProfiles": null,
  "controlPlaneProfile": {
    "kubernetesVersion": "1.31.9",
    "name": null,
    "osType": "Linux",
    "upgrades": [
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.5"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.4"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.3"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.2"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.1"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.32.0"
      }
    ]
  },
  "id": "/subscriptions/4af928af-144b-44ea-a5ad-9fe8d622bd21/resourcegroups/resource-aks/providers/Microsoft.ContainerService/managedClusters/akscluster/upgradeprofiles/default",
  "name": "default",
  "resourceGroup": "resource-aks",
  "type": "Microsoft.ContainerService/managedClusters/upgradeprofiles"
}
ayush [ ~ ]$ kubectl get nodes
NAME           STATUS  ROLES   AGE   VERSION
aks-nodepool1-18391766-vmss000001 Ready   <none>  46m   v1.31.9
aks-nodepool1-18391766-vmss000002 Ready   <none>  46m   v1.31.9
aks-nodepool1-18391766-vmss000003 Ready   <none>  16m   v1.31.9
aks-nodepool1-18391766-vmss000004 Ready   <none>  16m   v1.31.9
aks-nodepool1-18391766-vmss000005 Ready   <none>  16m   v1.31.9
ayush [ ~ ]$ az aks delete --resource-group resource-aks --name akscluster --yes --no-wait
ayush [ ~ ]$ az group delete --name resource-aks
Are you sure you want to perform this operation? (y/n): y

```

5. Configure liveness and readiness probes for pods in AKS cluster.

```
ayush [ ~ ]$ vim deployment-probe.yaml
ayush [ ~ ]$ cat deployment-probe.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: probe-demo
spec:
  replicas: 2
  selector:
    matchLabels:
      app: probe-demo
  template:
    metadata:
      labels:
        app: probe-demo
  spec:
    containers:
      - name: webapp
        image: nginx:latest
        ports:
          - containerPort: 80
    readinessProbe:
      httpGet:
        path: /
        port: 80
        initialDelaySeconds: 5
        periodSeconds: 5
        timeoutSeconds: 2
        failureThreshold: 3
    livenessProbe:
      httpGet:
        path: /
        port: 80
        initialDelaySeconds: 10
```

```

    initialDelaySeconds: 10
    periodSeconds: 10
    timeoutSeconds: 2
    failureThreshold: 3
ayush [ ~ ]$ kubectl apply -f deployment-probe.yaml
deployment.apps/probe-demo created
ayush [ ~ ]$ kubectl get pods -l app=probe-demo -w
NAME                      READY   STATUS    RESTARTS   AGE
probe-demo-6d4948c4b8-bmvrk   1/1     Running   0          37s
probe-demo-6d4948c4b8-knqbd   1/1     Running   0          37s

```

```

^Cayush [ ~ ]$ kubectl describe pod probe-demo
Name:           probe-demo-6d4948c4b8-bmvrk
Namespace:      default
Priority:       0
Service Account: default
Node:           aks-nodepool1-36362166-vms000001/10.224.0.5
Start Time:     Sun, 29 Jun 2025 20:35:52 +0000
Labels:         app=probe-demo
                pod-template-hash=6d4948c4b8
Annotations:    <none>
Status:         Running
IP:             10.244.0.239
IPs:
  IP:           10.244.0.239
Controlled By: ReplicaSet/probe-demo-6d4948c4b8
Containers:
  webapp:
    Container ID:  containerd://5f84841891390d94c103287fa69cadaa4a58667abd1ea46a5da883fa83c34f01
    Image:          nginx:latest
    Image ID:      docker.io/library/nginx@sha256:dc53c8f25a10f9109190ed5b59bda2d707a3bde0e45857ce9e1efaa32ff9cbc1
    Port:          80/TCP
    Host Port:    0/TCP
    State:         Running
      Started:   Sun, 29 Jun 2025 20:35:53 +0000
    Ready:         True
    Restart Count: 0
    Liveness:     http-get http://:80/ delay=10s timeout=2s period=10s #success=1 #failure=3
    Readiness:    http-get http://:80/ delay=5s timeout=2s period=5s #success=1 #failure=3
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-glnxp (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True

```

```

Ready:          True
Restart Count: 0
Liveness:      http-get http://:80/ delay=10s timeout=2s period=10s #success=1 #failure=3
Readiness:     http-get http://:80/ delay=5s timeout=2s period=5s #success=1 #failure=3
Environment:   <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-wftdd (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized       True
  Ready            True
  ContainersReady  True
  PodScheduled    True
Volumes:
  kube-api-access-wftdd:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    Optional:        false
    DownwardAPI:    true
  QoS Class:      BestEffort
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age From          Message
  ----  -----  --  --  -----
  Normal Scheduled  2m3s default-scheduler  Successfully assigned default/probe-demo-6d4948c4b8-knqbd to aks-nodepool1-36362166-vmss000001
  Normal Pulling   2m2s kubelet        Pulling image "nginx:latest"
  Normal Pulled    2m2s kubelet        Successfully pulled image "nginx:latest" in 344ms (344ms including waiting). Image size: 72225606 bytes.
  Normal Created   2m2s kubelet        Created container: webapp
  Normal Started   2m1s kubelet        Started container webapp

```

```

ayush [ ~ ]$ kubectl exec -it probe-demo-6d4948c4b8-bmvrk -- /bin/bash
root@probe-demo-6d4948c4b8-bmvrk:/# rm /etc/nginx/conf.d/default.conf
root@probe-demo-6d4948c4b8-bmvrk:/# exit
exit
ayush [ ~ ]$ kubectl get pods -w
NAME                           READY   STATUS    RESTARTS   AGE
probe-demo-6d4948c4b8-bmvrk   1/1     Running   0          6m29s
probe-demo-6d4948c4b8-knqbd   1/1     Running   0          6m29s
^Cayush [ ~ ]$ kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
probe-demo-6d4948c4b8-bmvrk   1/1     Running   0          6m43s
probe-demo-6d4948c4b8-knqbd   1/1     Running   0          6m43s
ayush [ ~ ]$ kubectl delete deployment probe-demo
deployment.apps "probe-demo" deleted
ayush [ ~ ]$ 

```

6. Configure Taints and Tolerants.

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
ayush [ ~ ]$ az group create --name akscluster --location eastus
{
  "id": "/subscriptions/4af928af-144b-44ea-a5ad-9fe8d622bd21/resourceGroups/akscluster",
  "location": "eastus",
  "managedBy": null,
  "name": "akscluster",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

```
ayush [ ~ ]$ az aks create --resource-group akscluster --name aks --node-count 3 --node-vm-size Standard_B2s --generate-ssh-key
SSH key files '/home/ayush/.ssh/id_rsa' and '/home/ayush/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If using machines without permanent storage like Azure Cloud Shell without an attached file share, back up your keys to a safe location
{
  "aadProfile": null,
  "addonProfiles": null,
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "capacityReservationGroupId": null,
      "count": 3,
      "creationData": null,
      "currentOrchestratorVersion": "1.31.9",
      "eTag": null,
      "enableAutoScaling": false,
      "enableEncryptionAtHost": false,
      "enableFips": false,
      "enableNodePublicIp": false,
      "enableUltraSsd": false,
      "gpuInstanceProfile": null,
      "gpuProfile": null,
      "hostGroupId": null,
      "kubeletConfig": null,
      "kubeletDiskType": "OS",
      "linuxOsConfig": null,
      "maxCount": null,
      "maxPods": 250,
      "messageOfTheDay": null,
      "minCount": null,
      "mode": "System",
      "name": "nodepool1",
      "networkProfile": null,
```

```

    "serviceMeshProfile": null,
    "servicePrincipalProfile": {
        "clientId": "msi",
        "secret": null
    },
    "sku": {
        "name": "Base",
        "tier": "Free"
    },
    "storageProfile": {
        "blobCsiDriver": null,
        "diskCsiDriver": {
            "enabled": true
        },
        "fileCsiDriver": {
            "enabled": true
        },
        "snapshotController": {
            "enabled": true
        }
    },
    "supportPlan": "KubernetesOfficial",
    "systemData": null,
    "tags": null,
    "type": "Microsoft.ContainerService/ManagedClusters",
    "upgradeSettings": null,
    "windowsProfile": null,
    "workloadAutoScalerProfile": {
        "keda": null,
        "verticalPodAutoscaler": null
    }
}
}

```

```

ayush [ ~ ]$ az aks get-credentials --resource-group akscluster --name aks
Merged "aks" as current context in /home/ayush/.kube/config
ayush [ ~ ]$ kubectl get nodes
NAME                  STATUS   ROLES      AGE     VERSION
aks-nodepool1-10059654-vmss000000  Ready    <none>    5m21s   v1.31.9
ays-nodepool1-10059654-vmss000001  Ready    <none>    5m7s    v1.31.9
aks-nodepool1-10059654-vmss000002  Ready    <none>    5m19s   v1.31.9
ayush [ ~ ]$ kubectl taint nodes aks-nodepool1 dedicated=frontend:NoSchedule
Error from server (NotFound): nodes "aks-nodepool1" not found
ayush [ ~ ]$ kubectl taint nodes aks-nodepool1-10059654-vmss000000 dedicated=frontend:NoSchedule
node/aks-nodepool1-10059654-vmss000000 tainted
ayush [ ~ ]$ kubectl describe node aks-nodepool1-10059654-vmss000000 | grep Taints
Taints:          dedicated=frontend:NoSchedule

```

```

ayush [ ~ ]$ vim frontend-pod.yaml
ayush [ ~ ]$ cat frontend-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: front-end-pod
spec:
  containers:
    - name: app
      image: nginx
  tolerations:
    - key: "dedicated"
      operator: "Equal"
      value: "frontend"
      effect: "NoSchedule"
ayush [ ~ ]$ kubectl apply -f frontend-pod.yaml
pod/front-end-pod created
ayush [ ~ ]$ kubectl get pods -owide
NAME           READY   STATUS    RESTARTS   AGE     IP           NODE          NOMINATED NODE   READINESS GATES
front-end-pod  1/1    Running   0          17s    10.244.0.82  aks-nodepool1-10059654-vmss000000  <none>        <none>
ayush [ ~ ]$ kubectl taint nodes aks-nodepool1-10059654-vmss000000 maintenance=planned:NoExecute
node/aks-nodepool1-10059654-vmss000000 tainted

```

```

ayush [ ~ ]$ vim deployment-main.yaml
ayush [ ~ ]$ cat deployment-main.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21
  tolerations:
    - key: "maintenance"
      operator: "planned"
      effect: "NoExecute"
      tolerationSeconds: 60
ayush [ ~ ]$ kubectl apply -f deployment-main.yaml
The Deployment "nginx" is invalid: spec.template.spec.tolerations[0].operator: Unsupported value: "planned": supported values: "Equal", "Exists"

```

```

ayush [ ~ ]$ vim deployment-main.yaml
ayush [ ~ ]$ cat deployment-main.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
  spec:
    containers:
      - name: nginx
        image: nginx:1.21
  tolerations:
    - key: "maintenance"
      operator: "Equal"
      value: "planned"
      effect: "NoExecute"
      tolerationSeconds: 60
ayush [ ~ ]$ kubectl apply -f deployment-main.yaml
deployment.apps/nginx created

```

```

ayush [ ~ ]$ kubectl taint nodes aks-nodepool1-10059654-vmss00000 maintenance=planned:NoExecute --overwrite
node/aks-nodepool1-10059654-vmss00000 modified
ayush [ ~ ]$ kubectl get pods -owide
NAME          READY   STATUS    RESTARTS   AGE     IP           NODE          NOMINATED NODE   READINESS GATES
nginx-5b8bb8c8c4-bh9m7  1/1   Running   0          3m59s  10.244.1.182  aks-nodepool1-10059654-vmss00002  <none>        <none>
nginx-5b8bb8c8c4-cdfhz  1/1   Running   0          3m59s  10.244.2.102  aks-nodepool1-10059654-vmss00001  <none>        <none>
ayush [ ~ ]$ kubectl describe pod nginx-5b8bb8c8c4-bh9m7 | grep -A3 Tolerations
Tolerations:      maintenance=planned:NoExecute for 60s
                  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
ayush [ ~ ]$ kubectl get pods -w
bash: kubectl: command not found
ayush [ ~ ]$ kubects get pods -w
bash: kubects: command not found
ayush [ ~ ]$ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
nginx-5b8bb8c8c4-bh9m7  1/1   Running   0          6m55s
nginx-5b8bb8c8c4-cdfhz  1/1   Running   0          6m55s

```

7. Create and attach persistent volume claims to pods.

```

ayush [ ~ ]$ az aks get-credentials --resource-group akscluster --name aks
Merged "aks" as current context in /home/ayush/.kube/config
ayush [ ~ ]$ vim pv.yaml
ayush [ ~ ]$ cat pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-volume
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: manual
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: "/mnt/data"
ayush [ ~ ]$ kubectl apply -f pv.yaml
persistentvolume/pv-volume created
ayush [ ~ ]$ kubectl get pv
NAME      CAPACITY   ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM     STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
pv-volume  1Gi        RWO          Retain          Available  manual    <unset>           8s

ayush [ ~ ]$ vim pvc.yaml
ayush [ ~ ]$ cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: manual
ayush [ ~ ]$ kubectl apply -f pvc.yaml
persistentvolumeclaim/pv-claim created
ayush [ ~ ]$ kubectl get pvc
NAME      STATUS  VOLUME      CAPACITY   ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
pv-claim  Bound   pv-volume  1Gi        RWO          manual       <unset>           13s
ayush [ ~ ]$ kubectl get pv pv-volume
NAME      CAPACITY   ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM     STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
pv-volume  1Gi        RWO          Retain          Available  default/pv-claim  manual       <unset>           4m18s
ayush [ ~ ]$ 

ayush [ ~ ]$ vim pod-pvc.yaml
ayush [ ~ ]$ cat pod-pvc.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pv-pod
spec:
  containers:
    - name: pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: pv-storage
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: pv-claim
ayush [ ~ ]$ kubectl apply -f pod-pvc.yaml
pod/pv-pod created
ayush [ ~ ]$ kubectl get pods -owide
NAME            READY   STATUS    RESTARTS   AGE     IP           NODE   NOMINATED NODE   READINESS GATES
nginx-5b8bb8c8c4-bh9m7  1/1    Running   0          175m   10.244.1.182  aks-nodepool1-10059654-vmss000002  <none>  <none>
nginx-5b8bb8c8c4-cdfhz  1/1    Running   0          175m   10.244.2.102  aks-nodepool1-10059654-vmss000001  <none>  <none>
pv-pod           1/1    Running   0          13s    10.244.1.170  aks-nodepool1-10059654-vmss000002  <none>  <none>

```

```

ayush [ ~ ]$ kubectl describe pod pv-pod
Name:          pv-pod
Namespace:     default
Priority:      0
Service Account: default
Node:          aks-nodepool1-10059654-vmss000002/10.224.0.4
Start Time:    Mon, 30 Jun 2025 09:18:46 +0000
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:           10.244.1.170
IPs:
  IP: 10.244.1.170
Containers:
  pv-container:
    Container ID: containerd://1101bdf76774c88fa2146816800c0af69a52d66066da56702dc41c0278968a1c
    Image:         nginx
    Image ID:     docker.io/library/nginx@sha256:dc53c8f25a10f9109190ed5b59bda2d707a3bde0e45857ce9e1efaa32ff9cbc1
    Port:          <none>
    Host Port:    <none>
    State:        Running
      Started:   Mon, 30 Jun 2025 09:18:50 +0000
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from pv-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tgsnl (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True

```

```

ContainersReady:      True
PodScheduled:        True
Volumes:
  pv-storage:
    Type:      PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  pv-claim
    ReadOnly:   false
  kube-api-access-tgsnl:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    Optional:    false
    DownwardAPI: true
  QoS Class:  BestEffort
  Node-Selectors:  <none>
  Tolerations:  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age  From      Message
  ----  -----  --  --       -----
  Normal Scheduled  30s default-scheduler  Successfully assigned default/pv-pod to aks-nodepool1-10059654-vmss000002
  Normal Pulling   30s kubelet        Pulling image "nginx"
  Normal Pulled    26s kubelet        Successfully pulled image "nginx" in 3.716s (3.716s including waiting). Image size: 72225606 bytes.
  Normal Created   26s kubelet        Created container: pv-container
  Normal Started   26s kubelet        Started container pv-container
ayush [ ~ ]$ kubectl exec -it pv-pod -- /bin/bash
root@pv-pod:/# cd /usr/share/nginx/html
root@pv-pod:/usr/share/nginx/html# echo "Hello Persistent Volume" > index.html
root@pv-pod:/usr/share/nginx/html# curl http://localhost
Hello Persistent Volume
root@pv-pod:/usr/share/nginx/html# exit
exit

```

```
ayush [ ~ ]$ vim deployment.yaml
ayush [ ~ ]$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          volumeMounts:
            - mountPath: /usr/share/nginx/html
              name: html-volume
      volumes:
        - name: html-volume
          persistentVolumeClaim:
            claimName: pv-claim
ayush [ ~ ]$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deploy created
ayush [ ~ ]$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
nginx        2/2     2           2           3h16m
nginx-deploy 2/2     2           2           15s
```

```

ayush [ ~ ]$ kubectl describe pods
Name:           nginx-5b8bb8c8c4-bh9m7
Namespace:      default
Priority:       0
Service Account: default
Node:           aks-nodepool1-10059654-vmss00002/10.224.0.4
Start Time:     Mon, 30 Jun 2025 06:23:00 +0000
Labels:         app=nginx
                pod-template-hash=5b8bb8c8c4
Annotations:    <none>
Status:         Running
IP:             10.244.1.182
IPs:
  IP:          10.244.1.182
Controlled By: ReplicaSet/nginx-5b8bb8c8c4
Containers:
  nginx:
    Container ID:  containerd://20e8f850eaac3d97f37db74cf24c4423cfa768b48eb9c790aa5eee9675dd5cb8
    Image:         nginx:1.21
    Image ID:     docker.io/library/nginx@sha256:2bcabc23b45489fb0885d69a06ba1d648aeda973fae7bb981bafbb884165e514
    Port:          <none>
    Host Port:    <none>
    State:        Running
      Started:   Mon, 30 Jun 2025 06:23:05 +0000
    Ready:         True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-m5nhw (ro)
Conditions:
  Type          Status
  PodReadyToStartContainers  True

```

```

Ready          True
ContainersReady  True
PodScheduled    True
Volumes:
pv-storage:
  Type:      PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
  ClaimName: pv-claim
  ReadOnly:   false
  kube-api-access-tgsnl:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    Optional:    false
    DownwardAPI: true
  QoS Class:  BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age   From            Message
  ----  -----  --   --              --
  Normal Scheduled  20m  default-scheduler  Successfully assigned default/pv-pod to aks-nodepool1-10059654-vmss00002
  Normal Pulling   20m  kubelet         Pulling image "nginx"
  Normal Pulled    20m  kubelet         Successfully pulled image "nginx" in 3.716s (3.716s including waiting). Image size: 72225606 bytes.
  Normal Created   20m  kubelet         Created container: pv-container
  Normal Started   20m  kubelet         Started container pv-container
ayush [ ~ ]$ kubectl get pv,pvc
NAME           CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM           STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
persistentvolume/pv-volume  1Gi        RWO          Retain        Bound    default/pv-claim  manual        <unset>          28m
NAME           STATUS    VOLUME      CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/pv-claim  Bound    pv-volume   1Gi        RWO          manual        <unset>          24m

```

```
ayush [ ~ ]$ kubectl delete pod -l app=nginx
pod "nginx-5b8bb8c8c4-bh9m7" deleted
pod "nginx-5b8bb8c8c4-cdfhz" deleted
pod "nginx-deploy-7f77fc56f5-567g2" deleted
pod "nginx-deploy-7f77fc56f5-772t9" deleted
```

```
ayush [ ~ ]$ kubectl delete pvc pv-claim
persistentvolumeclaim "pv-claim" deleted
```

```
^Cayush [ ~ ]$ kubectl delete pv pv-volume
persistentvolume "pv-volume" deleted
```

8. Configure health probes for pods.

```
ayush [ ~ ]$ vim frontend-pod.yaml
ayush [ ~ ]$ cat frontend-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: health-probe
spec:
  containers:
    - name: app
      image: nginx:latest
      ports:
        - containerPort: 80
      readinessProbe:
        httpGet:
          path: /
          port: 80
          initialDelaySeconds: 5
          periodSeconds: 5
          timeoutSeconds: 2
          successThreshold: 1
          failureThreshold: 3
      livenessProbe:
        httpGet:
          path: /
          port: 80
          initialDelaySeconds: 10
          periodSeconds: 10
          timeoutSeconds: 2
          failureThreshold: 3
      startupProbe:
        httpGet:
          path: /
```

```

path: /
port: 80
initialDelaySeconds: 20
periodSeconds: 10
failureThreshold: 30
ayush [ ~ ]$ kubectl apply -f frontend-pod.yaml
pod/health-probe created
ayush [ ~ ]$ kubectl get pods -owide
NAME          READY   STATUS    RESTARTS   AGE     IP           NODE
health-probe  0/1     Running   0          11s    10.244.2.45  aks-nodepool1-10059654-vmss000001
nginx-5b8bb8c8c4-dt5js  1/1     Running   0          46m    10.244.1.42  aks-nodepool1-10059654-vmss000002
nginx-5b8bb8c8c4-ml9cl  1/1     Running   0          46m    10.244.2.141 aks-nodepool1-10059654-vmss000001
nginx-deploy-7f77fc56f5-nz754  1/1     Running   0          46m    10.244.1.32  aks-nodepool1-10059654-vmss000002
nginx-deploy-7f77fc56f5-xdjm7  1/1     Running   0          46m    10.244.2.162 aks-nodepool1-10059654-vmss000001
pv-pod         1/1     Running   0          68m    10.244.1.170 aks-nodepool1-10059654-vmss000002
ayush [ ~ ]$ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
health-probe  0/1     Running   0          24s
nginx-5b8bb8c8c4-dt5js  1/1     Running   0          46m
nginx-5b8bb8c8c4-ml9cl  1/1     Running   0          46m
nginx-deploy-7f77fc56f5-nz754  1/1     Running   0          46m
nginx-deploy-7f77fc56f5-xdjm7  1/1     Running   0          46m
pv-pod         1/1     Running   0          68m
health-probe  0/1     Running   0          30s
health-probe  1/1     Running   0          30s
^Cayush [ ~ ]$
```

```

ayush [ ~ ]$ kubectl describe pod health-probe
Name:           health-probe
Namespace:      default
Priority:       0
Service Account: default
Node:           aks-nodepool1-10059654-vmss000001/10.224.0.6
Start Time:     Mon, 30 Jun 2025 10:27:07 +0000
Labels:         <none>
Annotations:   <none>
Status:         Running
IP:            10.244.2.45
IPs:
  IP: 10.244.2.45
Containers:
  app:
    Container ID: containerd://de1aaefe4cc52def9f615ea3fc2ddd279a88f26fa7c9e6e4977dfa04f145ee96
    Image:        nginx:latest
    Image ID:    docker.io/library/nginx@sha256:dc53c8f25a10f9109190ed5b59bda2d707a3bde0e45857ce9e1efaa32ff9cbc1
    Port:         80/TCP
    Host Port:   0/TCP
    State:        Running
      Started:   Mon, 30 Jun 2025 10:27:08 +0000
    Ready:        True
    Restart Count: 0
    Liveness:    http-get http://:80/ delay=10s timeout=2s period=10s #success=1 #failure=3
    Readiness:   http-get http://:80/ delay=5s timeout=2s period=5s #success=1 #failure=3
    Startup:    http-get http://:80/ delay=20s timeout=1s period=10s #success=1 #failure=30
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-vj2fm (ro)
Conditions:
  Type  Status
```

```

PodReadyToStartContainers: True
Initialized: True
Ready: True
ContainersReady: True
PodScheduled: True
Volumes:
  kube-api-access-vj2fm:
    Type:            Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:     kube-root-ca.crt
    Optional:         false
    DownwardAPI:      true
  QoS Class:        BestEffort
  Node-Selectors:   <none>
  Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age   From          Message
  ----  -----  --  --  -----
  Normal Scheduled  111s default-scheduler  Successfully assigned default/health-probe to aks-nodepool1-10059654-vmss000001
  Normal Pulling   110s kubelet        Pulling image "nginx:latest"
  Normal Pulled    110s kubelet        Successfully pulled image "nginx:latest" in 217ms (217ms including waiting). Image size: 72225606 bytes.
  Normal Created   110s kubelet        Created container: app
  Normal Started   110s kubelet        Started container app
ayush [ ~ ]$ kubectl get pods -w
NAME                  READY   STATUS    RESTARTS   AGE
health-probe          1/1     Running   0          2m15s
nginx-5b8bb8c8c4-dt5js 1/1     Running   0          48m
nginx-5b8bb8c8c4-m19cl 1/1     Running   0          48m
nginx-deploy-7f77fc56f5-nz754 1/1     Running   0          48m
nginx-deploy-7f77fc56f5-xdjm7 1/1     Running   0          48m
pv-pod                1/1     Running   0          70m

```

9. Configure autoscaling in your cluster (Horizontal scaling)

```

ayush [ ~ ]$ vim deployment.yaml
ayush [ ~ ]$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          resources:
            requests:
              cpu: 200m
            limits:
              cpu: 500m
          volumeMounts:
            - mountPath: /usr/share/nginx/html
              name: html-volume
      volumes:
        - name: html-volume
      persistentVolumeClaim:
        claimName: pv-claim

```

```

ayush [ ~ ]$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deploy created
ayush [ ~ ]$ vim hpa.yaml
ayush [ ~ ]$ cat hpa.yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deploy
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 50
    - type: Resource
      resource:
        name: memory
        target:
          type: Utilization
          averageUtilization: 70
ayush [ ~ ]$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/nginx-hpa created

```

```

ayush [ ~ ]$ kubectl describe hpa nginx-hpa
Name:                   nginx-hpa
Namespace:              default
Labels:                 <none>
Annotations:            <none>
CreationTimestamp:      Mon, 30 Jun 2025 11:08:41 +0000
Reference:              Deployment/nginx-deploy
Metrics:                ( current / target )
  resource cpu on pods  (as a percentage of request): <unknown> / 50%
  resource memory on pods  (as a percentage of request): <unknown> / 70%
Min replicas:           1
Max replicas:           10
Deployment pods:        0 current / 0 desired
Events:                 <none>
ayush [ ~ ]$ kubectl get hpa
NAME      REFERENCE          TARGETS          MINPODS   MAXPODS   REPLICAS
AGE
nginx-hpa  Deployment/nginx-deploy  cpu: <unknown>/50%, memory: <unknown>/70%  1         10        2
25s

```

```
ayush [ ~ ]$ kubectl run -i --tty load-generator --rm --image=busybox --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://nginx; done"
If you don't see a command prompt, try pressing enter.
wget: bad address 'nginx'
```

```
ayush [ ~ ]$ kubectl get hpa nginx-hpa -w
NAME      REFERENCE          TARGETS          MINPODS   MAXPODS   REPLICAS
AGE
nginx-hpa  Deployment/nginx-deploy  cpu: <unknown>/50%, memory: <unknown>/70%  1          10         2
2m18s
^Z
[1]+  Stopped                  kubectl get hpa nginx-hpa -w
ayush [ ~ ]$ kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
aks-nodepool1-10059654-vmss000000  Ready   <none>  5h23m  v1.31.9
aks-nodepool1-10059654-vmss000001  Ready   <none>  5h23m  v1.31.9
aks-nodepool1-10059654-vmss000002  Ready   <none>  5h23m  v1.31.9
```

```
ayush [ ~ ]$ az aks nodepool update --resource-group akscluster --cluster-name aks --name nodepool1 --enable-cluster-autoscaler --min-count 1 --max-count 5
{
  "availabilityZones": null,
  "capacityReservationGroupId": null,
  "count": 3,
  "creationData": null,
  "currentOrchestratorVersion": "1.31.9",
  "eTag": null,
  "enableAutoScaling": true,
  "enableEncryptionAtHost": false,
  "enableIps": false,
  "enableNodePublicIp": false,
  "enableUltraSsd": false,
  "gpuInstanceProfile": null,
  "gpuProfile": null,
  "hostGroupId": null,
  "id": "/subscriptions/4af928af-144b-44ea-a5ad-9fe8d622bd21/resourcegroups/akscluster/providers/Microsoft.ContainerService/managedClusters/aks/agentPools/nodepool1",
  "kubeletConfig": null,
  "kubeletDiskType": "OS",
  "linuxOsConfig": null,
  "maxCount": 5,
  "maxPods": 250,
  "messageOfTheDay": null,
  "minCount": 1,
  "mode": "System",
  "name": "nodepool1",
  "networkProfile": null,
  "nodeImageVersion": "AKSUbuntu-2204gen2containerd-202506.16.0",
  "nodeLabels": null,
  "nodePublicIpPrefixId": null,
  "nodeTaints": null,
  "orchestratorVersion": "1.31",
}

"osSku": "Ubuntu",
"osType": "Linux",
"podSubnetId": null,
"powerState": {
  "code": "Running"
},
"provisioningState": "Succeeded",
"proximityPlacementGroupId": null,
"resourceGroup": "akscluster",
"scaleDownMode": "Delete",
"scaleSetEvictionPolicy": null,
"scaleSetPriority": null,
"securityProfile": {
  "enableSecureBoot": false,
  "enableVtpm": false
},
"spotMaxPrice": null,
"tags": null,
"type": "Microsoft.ContainerService/managedClusters/agentPools",
"typePropertiesType": "VirtualMachineScaleSets",
"upgradeSettings": {
  "drainTimeoutInMinutes": null,
  "maxSurge": "10%",
  "maxUnavailable": "0",
  "nodeSoakDurationInMinutes": null,
  "undrainableNodeBehavior": null
},
"vmSize": "Standard_B2s",
"vnetSubnetId": null,
"windowsProfile": null,
"workloadRuntime": null
}
```

```
ayush [ ~ ]$ kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
aks-nodepool1-10059654-vmss000000  Ready   <none>  5h32m  v1.31.9
aks-nodepool1-10059654-vmss000001  Ready   <none>  5h31m  v1.31.9
aks-nodepool1-10059654-vmss000002  Ready   <none>  5h31m  v1.31.9
ayush [ ~ ]$ az aks nodepool show --resource-group akscluster --cluster-name aks --name nodepool1
{
  "availabilityZones": null,
  "capacityReservationGroupId": null,
  "count": 3,
  "creationData": null,
  "currentOrchestratorVersion": "1.31.9",
  "eTag": null,
  "enableAutoScaling": true,
  "enableEncryptionAtHost": false,
  "enableFips": false,
  "enableNodePublicIp": false,
  "enableUltraSsd": false,
  "gpuInstanceProfile": null,
  "gpuProfile": null,
  "hostGroupId": null,
  "id": "/subscriptions/4af928af-144b-44ea-a5ad-9fe8d622bd21/resourcegroups/akscluster/providers/Microsoft.ContainerService/managedClusters/aks/agentPools/nodepool1",
  "kubeletConfig": null,
  "kubeletDiskType": "OS",
  "linuxOsConfig": null,
  "maxCount": 5,
  "maxPods": 250,
  "messageOfTheDay": null,
  "minCount": 1,
  "mode": "System",
  "name": "nodepool1",
  "networkProfile": null,
  "osType": "Linux",
  "podSubnetId": null,
  "powerState": {
    "code": "Running"
  },
  "provisioningState": "Succeeded",
  "proximityPlacementGroupId": null,
  "resourceGroup": "akscluster",
  "scaleDownMode": "Delete",
  "scaleSetEvictionPolicy": null,
  "scaleSetPriority": null,
  "securityProfile": {
    "enableSecureBoot": false,
    "enableVtpm": false
  },
  "spotMaxPrice": null,
  "tags": null,
  "type": "Microsoft.ContainerService/managedClusters/agentPools",
  "typePropertiesType": "VirtualMachineScaleSets",
  "upgradeSettings": {
    "drainTimeoutInMinutes": null,
    "maxSurge": "10%",
    "maxUnavailable": "0",
    "nodeSoakDurationInMinutes": null,
    "undrainableNodeBehavior": null
  },
  "vmSize": "Standard_B2s",
  "vnetSubnetId": null,
  "windowsProfile": null,
  "workloadRuntime": null
}
ayush [ ~ ]$
```