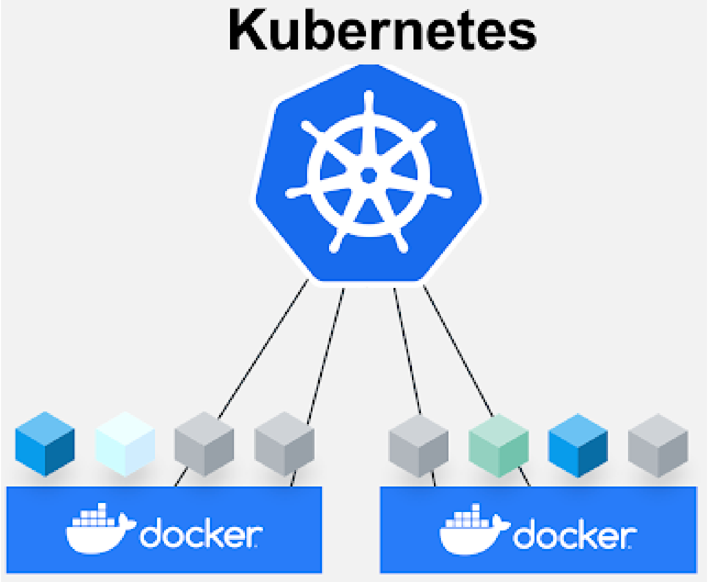
## 9.1 Build a Docker Application with Kubernetes Service (Example 1)

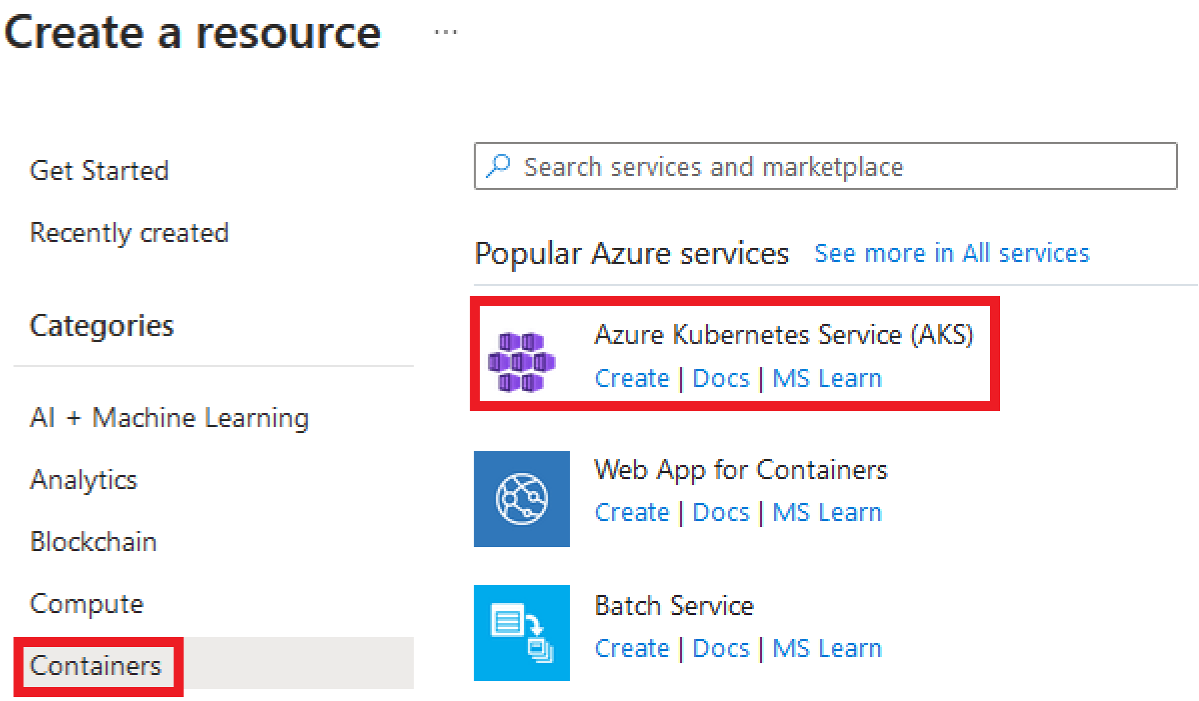


**Objectives:**

- Create a Kubernetes cluster.
- Build a Docker application with azure Kubernetes service.

**Part 1. Create a Kubernetes Cluster**

1. From Aure Portal, create a new resource. Select Containers➔ AKS.



Create a resource ···

Get Started

Recently created

Categories

AI + Machine Learning

Analytics

Blockchain

Compute

Containers

Search services and marketplace

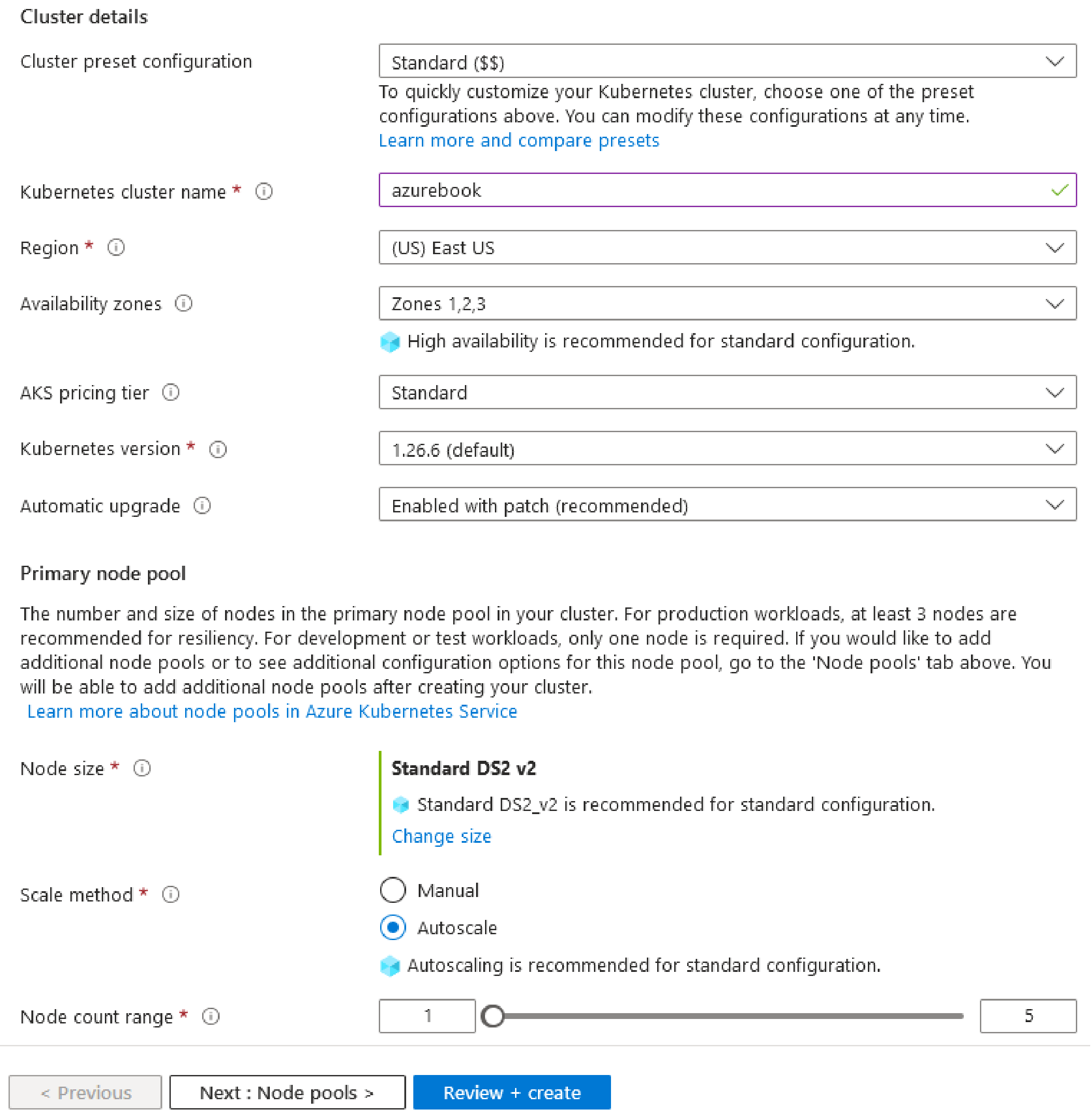
Popular Azure services See more in All services

Azure Kubernetes Service (AKS)
Create | Docs | MS Learn

Web App for Containers
Create | Docs | MS Learn

Batch Service
Create | Docs | MS Learn

2. Assign cluster a name and use default settings, including auto scale. Click “**Review + create**”.



Cluster details

| | |
|---|---|
| Cluster preset configuration | Standard ($$)<br>To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.<br>Learn more and compare presets |
| Kubernetes cluster name * ⓘ | azurebook ✓ |
| Region * ⓘ | (US) East US |
| Availability zones ⓘ | Zones 1,2,3<br>High availability is recommended for standard configuration. |
| AKS pricing tier ⓘ | Standard |
| Kubernetes version * ⓘ | 1.26.6 (default) |
| Automatic upgrade ⓘ | Enabled with patch (recommended) |

Primary node pool

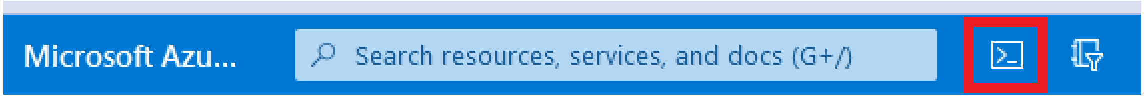
The number and size of nodes in the primary node pool in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. If you would like to add additional node pools or to see additional configuration options for this node pool, go to the 'Node pools' tab above. You will be able to add additional node pools after creating your cluster.
Learn more about node pools in Azure Kubernetes Service

| | |
|---|---|
| Node size * ⓘ | **Standard DS2 v2**<br>Standard DS2_v2 is recommended for standard configuration.<br>Change size |
| Scale method * ⓘ | ○ Manual<br>● Autoscale<br>Autoscaling is recommended for standard configuration. |
| Node count range * ⓘ | 1 – 5 |

3. Click “Create”.
4. Go to resource when deployment is complete.

## Part 2. Connect to the Cluster

5. Open Cloud Shell using the button on the top of the Azure portal.



6. Connect to your Kubernetes cluster using the az aks get-credentials command. Replace **myResourceGroup** and **myAKSCluster** with your own values.

```
$ az aks get-credentials --resource-group myResourceGroup --name myAKSCluster
```

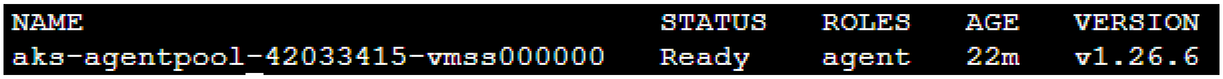
7. In my case the command is the following.

```
$ az aks get-credentials --resource-group azurebook --name azurebook
```

8. Verify the connection to your cluster using the following command.

```
$ kubectl get nodes
```

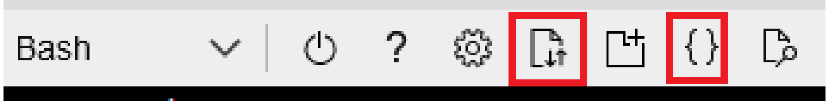
9. This is a sample output. Make sure the node status is Ready.



```
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-42033415-vmss000000   Ready    agent   22m   v1.26.6
```

## Part 3. Deploy the Application

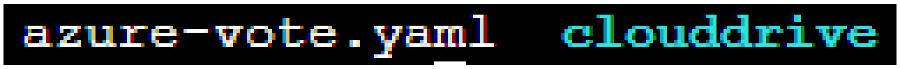
10. In the Cloud Shell, upload file **azure-vote.yaml** given online or use the text editor to create this file.



11. Use command **ls** to show this file.

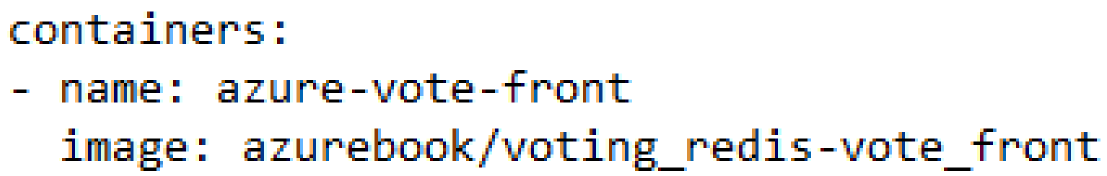
```
$ ls
```

12. This is the output.



```
azure-vote.yaml  clouddrive
```

13. Note, in file **azure-vote.yaml** , the front end image was created in previous chapters and is stored in Docker hub.

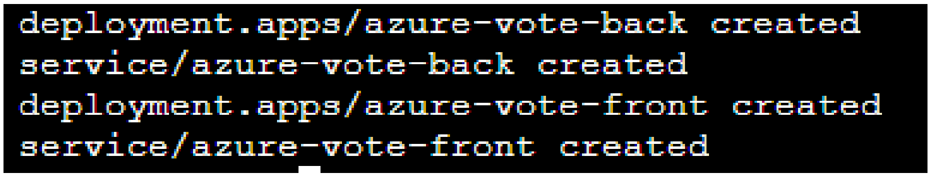


```
containers:
- name: azure-vote-front
  image: azurebook/voting_redis-vote_front
```

14. Deploy the application.

**$ kubectl apply -f azure-vote.yaml**

15. Output is like the following.

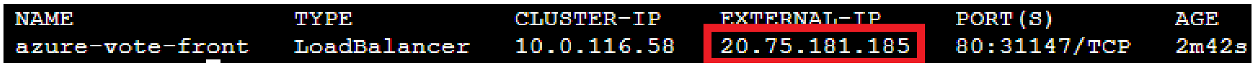


```
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
```

16. Use the following command to monitor the service.

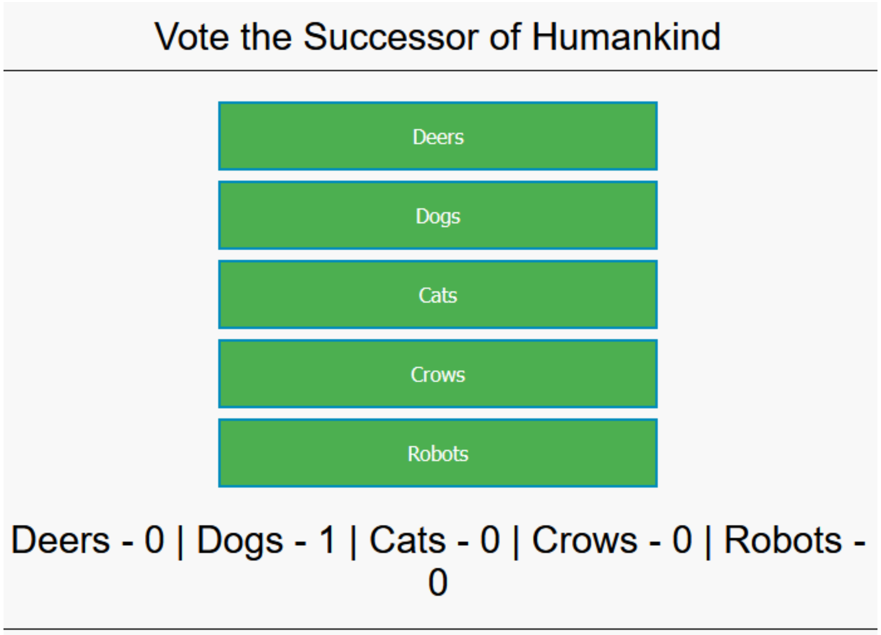
**$ kubectl get service azure-vote-front**

17. This is the output.



```
NAME              TYPE          CLUSTER-IP   EXTERNAL-IP    PORT(S)       AGE
azure-vote-front  LoadBalancer  10.0.116.58  20.75.181.185  80:31147/TCP  2m42s
```

18. To test your application, open a web browser to the external IP address of your service shown at Step 17.



Vote the Successor of Humankind

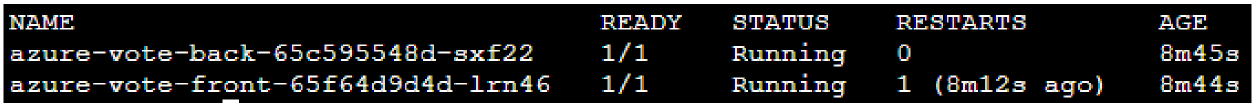
Deers - 0 | Dogs - 1 | Cats - 0 | Crows - 0 | Robots - 0

## Part 4. Manually Scale Pods

19. When you deployed the application, a single replica was created for both the frontend and the backend services. To see the number and state of pods in your cluster, use the following command.

```
$ kubectl get pods
```

20. The output looks like the following.



```
NAME                                 READY   STATUS    RESTARTS        AGE
azure-vote-back-65c595548d-sxf22     1/1     Running   0               8m45s
azure-vote-front-65f64d9d4d-lrn46    1/1     Running   1 (8m12s ago)   8m44s
```

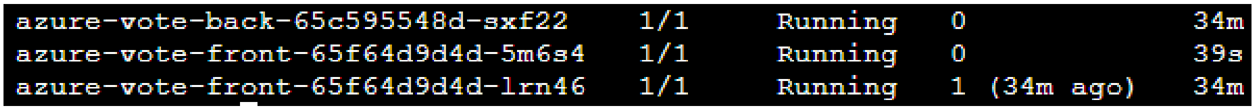
21. Manually change the number of pods in the azure-vote-front deployment using the **kubectl scale** command. The following example command increases the number of front-end pods to two.

```
$ kubectl scale --replicas=2 deployment/azure-vote-front
```

22. Verify the additional pods were created.

```
$ kubectl get pods
```

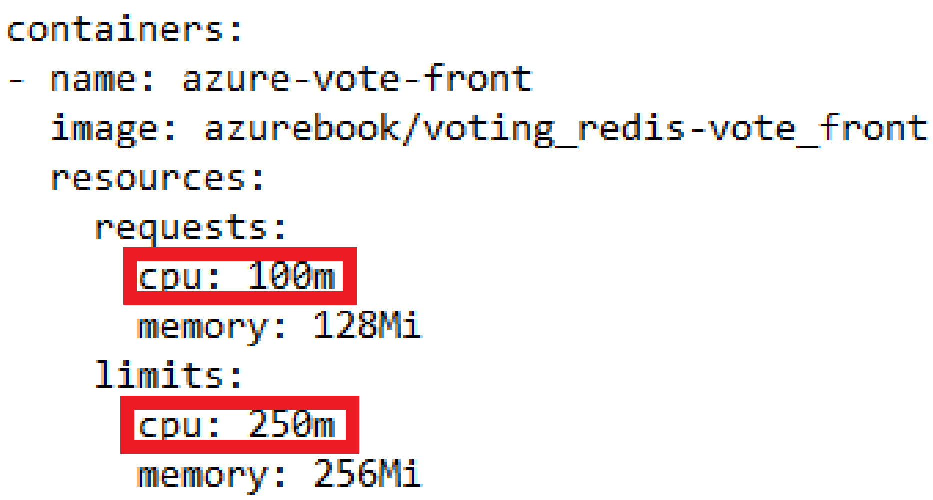
23. The output looks like the following.



```
azure-vote-back-65c595548d-sxf22     1/1     Running   0              34m
azure-vote-front-65f64d9d4d-5m6s4    1/1     Running   0              39s
azure-vote-front-65f64d9d4d-lrn46    1/1     Running   1 (34m ago)    34m
```

## Part 5. Auto Scale Pods

24. Kubernetes supports autoscaling to adjust the number of pods in a deployment depending on CPU utilization or other metrics.
25. To use the autoscaler, all containers and pods must have defined CPU requests and limits. In the azure-vote-front deployment, the front-end container requests 0.1 CPU with a limit of 0.25 CPU as shown below.



```
containers:
- name: azure-vote-front
  image: azurebook/voting_redis-vote_front
  resources:
    requests:
      cpu: 100m
      memory: 128Mi
    limits:
      cpu: 250m
      memory: 256Mi
```

26. The following command autoscales the number of pods in the azure-vote-front deployment with the following conditions: if average CPU utilization across all pods exceeds 50% of the requested usage, the autoscaler increases the pods up to a maximum of 5 instances and a minimum of 3 instances for the deployment.

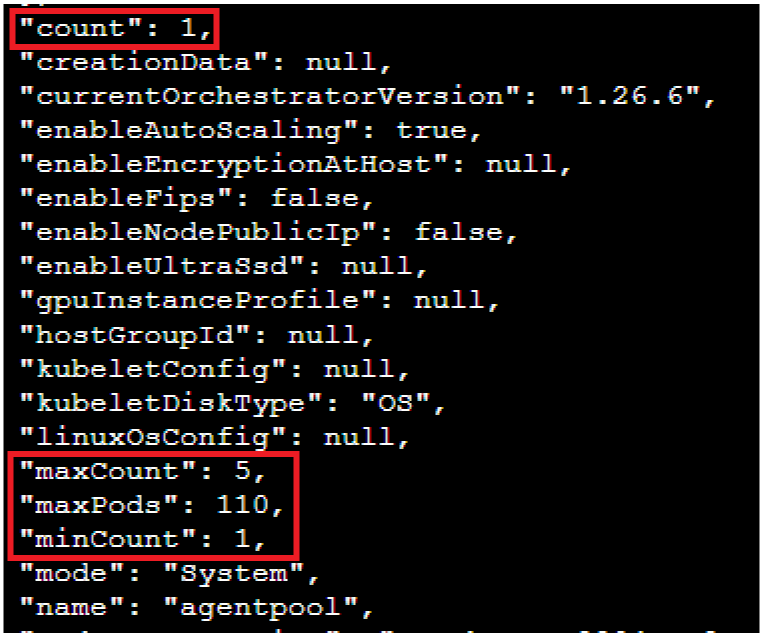
    **$ kubectl autoscale deployment azure-vote-front --cpu-percent=50 --min=3 --max=5**

## Part 6. Scale the Cluster Nodes

27. Get the number of nodes using the **az aks show** command.

    **$ az aks show --resource-group azurebook --name azurebook --query agentPoolProfiles**

28. It shows the following info: one node is deployed. It can be scaled between 1 and 5.



```
"count": 1,
"creationData": null,
"currentOrchestratorVersion": "1.26.6",
"enableAutoScaling": true,
"enableEncryptionAtHost": null,
"enableFips": false,
"enableNodePublicIp": false,
"enableUltraSsd": null,
"gpuInstanceProfile": null,
"hostGroupId": null,
"kubeletConfig": null,
"kubeletDiskType": "OS",
"linuxOsConfig": null,
"maxCount": 5,
"maxPods": 110,
"minCount": 1,
"mode": "System",
"name": "agentpool",
```

29. Because auto scale is selected when we create azure Kubernetes service, we cannot manually scale the cluster.
30. If manual is selected when we create azure Kubernetes service, we can scale our cluster nodes using the az aks scale command.

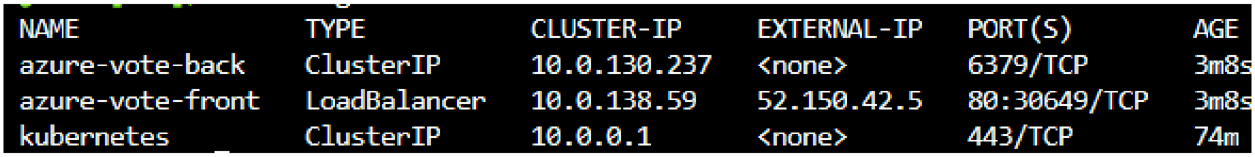
```
$ az aks scale --resource-group azurebook --name azurebook --node-count 3
```

## Part 8. Delete Service and Restart the Services

31. Use the following command to list all services.

```
$ kubectl get service
```

32. The output is like following.



```
NAME              TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
azure-vote-back   ClusterIP      10.0.130.237   <none>        6379/TCP       3m8s
azure-vote-front  LoadBalancer   10.0.138.59    52.150.42.5   80:30649/TCP   3m8s
kubernetes        ClusterIP      10.0.0.1       <none>        443/TCP        74m
```

33. Use the following command to delete the front service.

```
$ kubectl delete service azure-vote-front
```

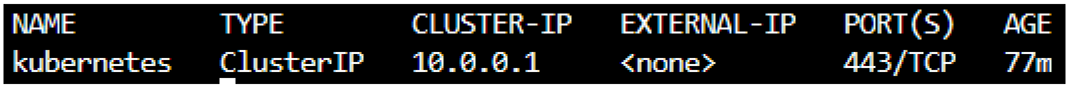
34. Use the following command to delete the backend service.

```
$ kubectl delete service azure-vote-back
```

35. List all services.

```
$ kubectl get service
```

36. The output is like the following.



```
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.0.0.1     <none>        443/TCP   77m
```

37. Use the following command to restart the application.

```
$ kubectl apply -f azure-vote.yaml
```

38. Test the application in a web browser.
39. Delete all resources from Azure Portal.