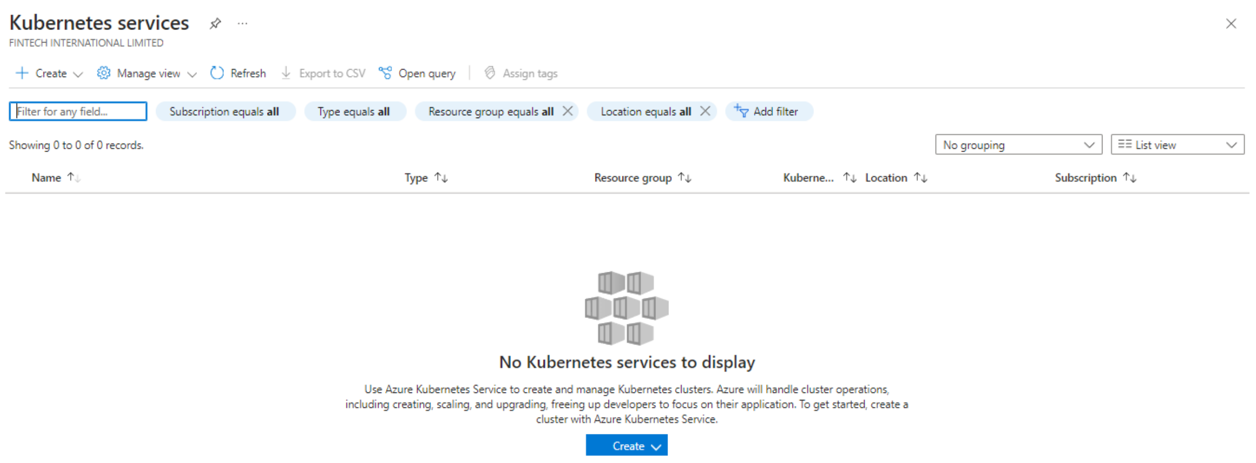
DEPLOY AN AZURE KUBERNETES SERVICE CLUSTER

We need to understand a few terms as far as kubernetes is concerned;

The virtual machine where our aks is running is called **node**

Containers will be running inside **Pods** are the smallest unit of AKS and is where we do our deployment

1) Log into the Azure Portal and under "Search" type kubernetes services



Kubernetes services
FINTECH INTERNATIONAL LIMITED

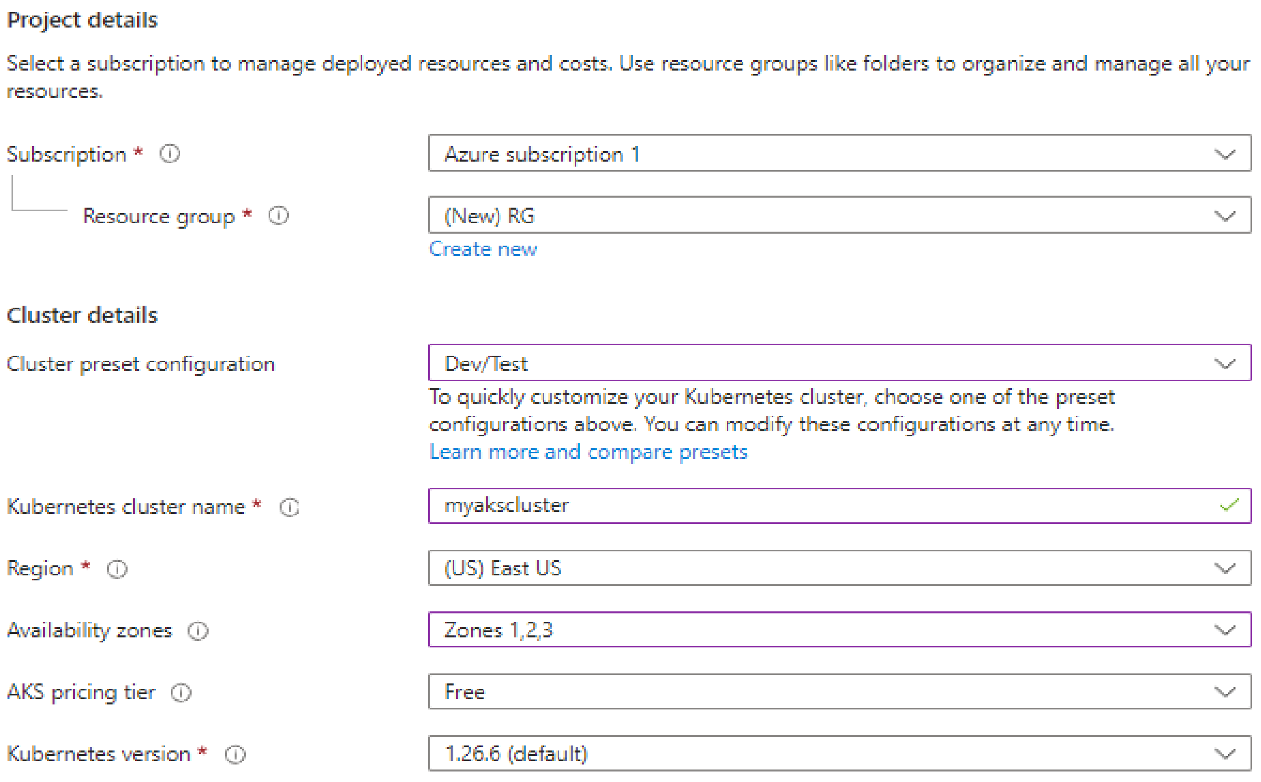
Showing 0 to 0 of 0 records.

Name	Type	Resource group	Kuberne...	Location	Subscription

No Kubernetes services to display

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

2) Click on create and on the Basics tab, specify resource group, name to your cluster, the region you want to create your cluster, availability zone, kubernetes version, Node size, node count value i.e. how many worker nodes we want.



Project details

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

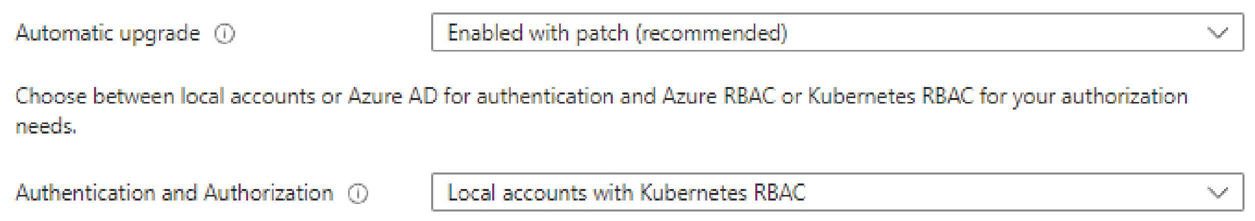
Subscription *	Azure subscription 1
Resource group *	(New) RG

Cluster details

Cluster preset configuration	Dev/Test

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

Kubernetes cluster name *	myakscluster
Region *	(US) East US
Availability zones	Zones 1,2,3
AKS pricing tier	Free
Kubernetes version *	1.26.6 (default)

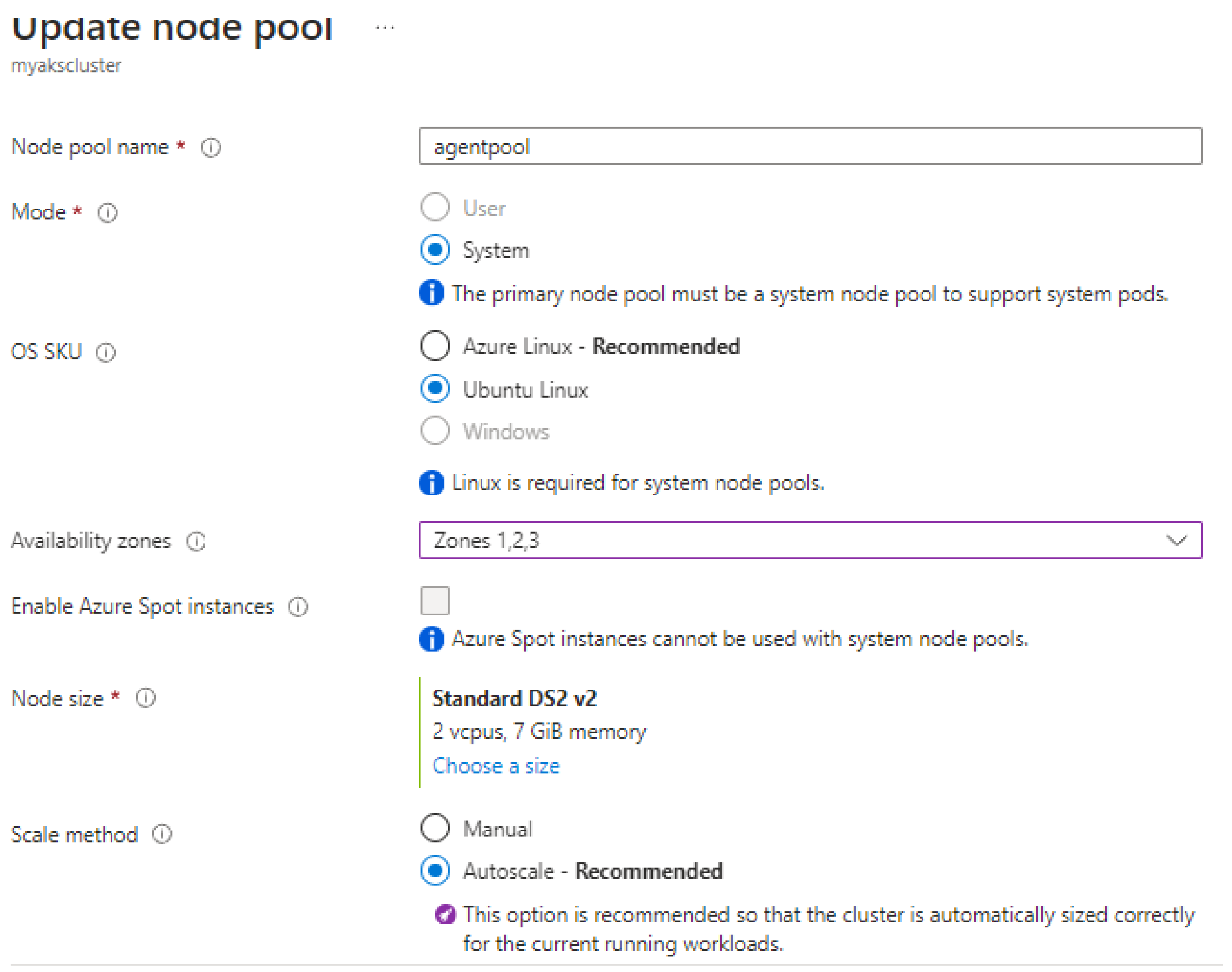


Automatic upgrade ⓘ	Enabled with patch (recommended)

Choose between local accounts or Azure AD for authentication and Azure RBAC or Kubernetes RBAC for your authorization needs.

Authentication and Authorization ⓘ	Local accounts with Kubernetes RBAC

3) Modify the node pool as per your preference i.e. Availability zones, node size, scale method, minimum node count, maximum node count. I have left mine with the default values as it is a test environment. Once you are done click on the update button

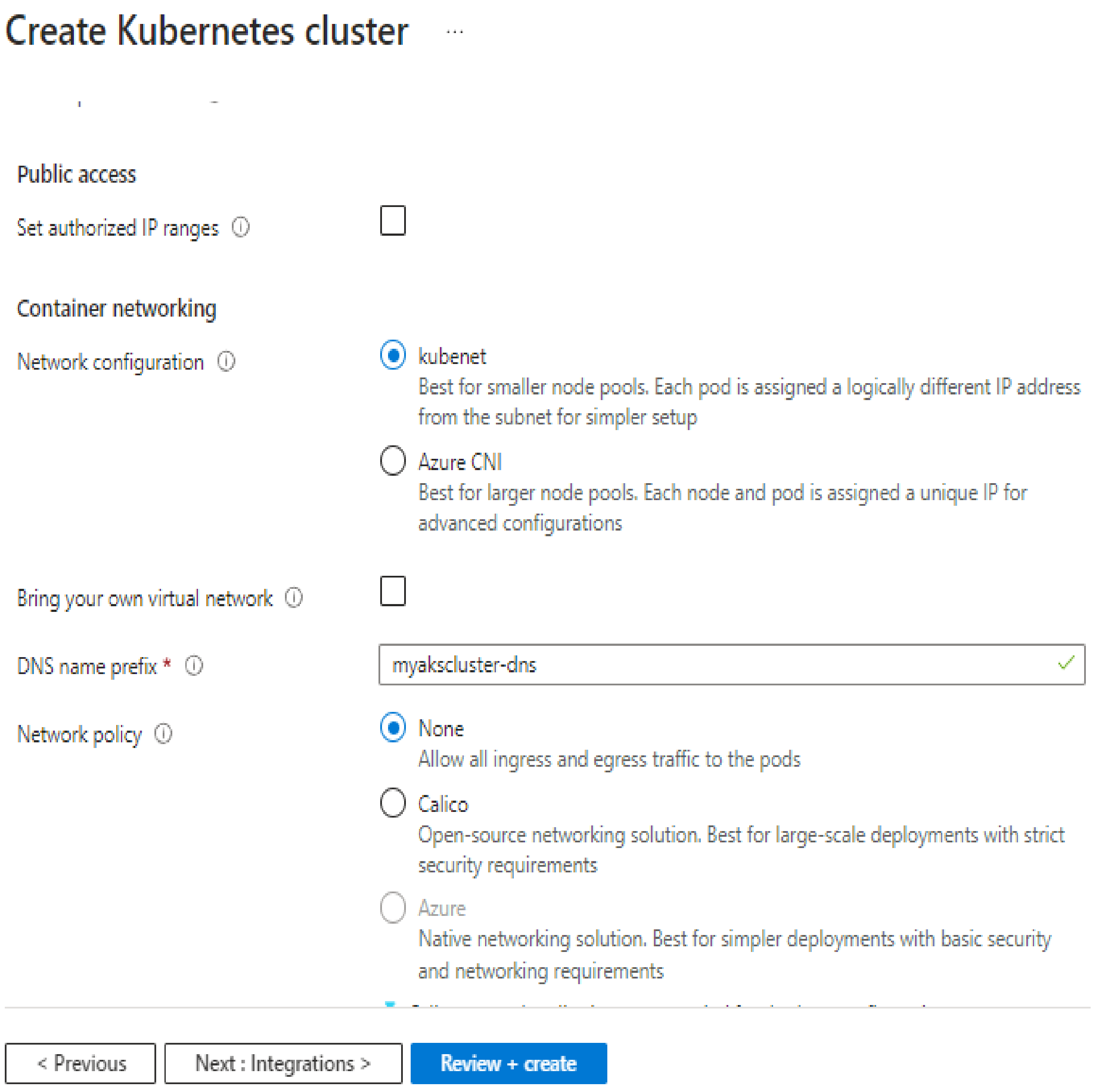


Update node pool ...

myakscluster

Node pool name * ⓘ	agentpool
Mode * ⓘ	User [✓ System] The primary node pool must be a system node pool to support system pods.
OS SKU ⓘ	Azure Linux - **Recommended** [✓ Ubuntu Linux] Windows Linux is required for system node pools.
Availability zones ⓘ	Zones 1,2,3
Enable Azure Spot instances ⓘ	☐ Azure Spot instances cannot be used with system node pools.
Node size * ⓘ	**Standard DS2 v2** 2 vcpus, 7 GiB memory Choose a size
Scale method ⓘ	Manual [✓ Autoscale - **Recommended**] This option is recommended so that the cluster is automatically sized correctly for the current running workloads.

4) Next we'll proceed to the networking tab. Here I want Microsoft to be responsible for my networking so I'll go ahead and click kubenet under "Network configuration". Further because we want the pods to communicate to each other I'll choose None under "Network policy"


Create Kubernetes cluster

Public access

Set authorized IP ranges ☐

Container networking

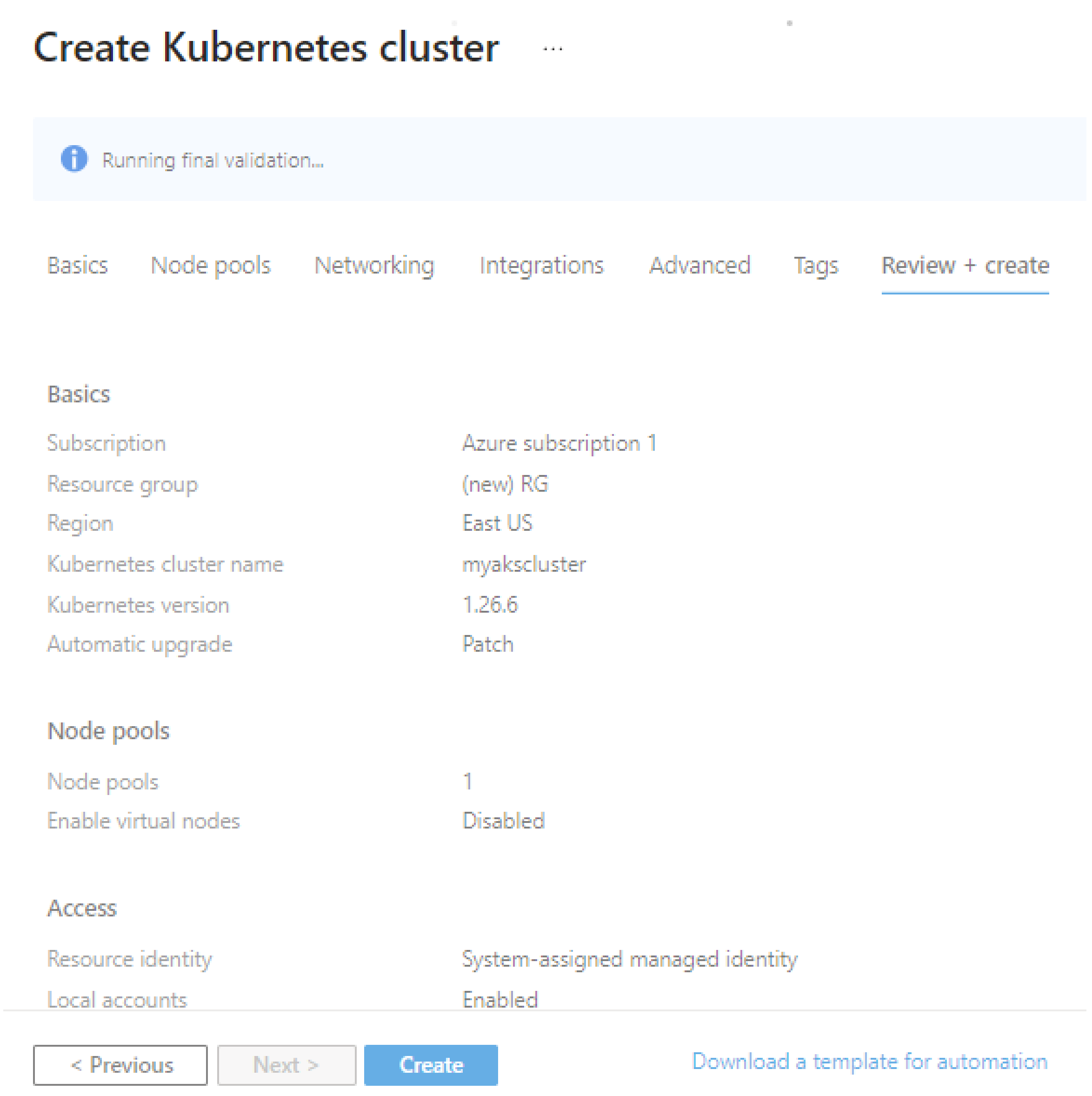
Network configuration [✓ kubenet]
Best for smaller node pools. Each pod is assigned a logically different IP address from the subnet for simpler setup
Azure CNI
Best for larger node pools. Each node and pod is assigned a unique IP for advanced configurations

Bring your own virtual network ☐

DNS name prefix * myakscluster-dns ✓

Network policy [✓ None]
Allow all ingress and egress traffic to the pods
Calico
Open-source networking solution. Best for large-scale deployments with strict security requirements
Azure
Native networking solution. Best for simpler deployments with basic security and networking requirements

5) Click on review and create



Create Kubernetes cluster

Running final validation...

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

Basics

Subscription	Azure subscription 1
Resource group	(new) RG
Region	East US
Kubernetes cluster name	myakscluster
Kubernetes version	1.26.6
Automatic upgrade	Patch

Node pools

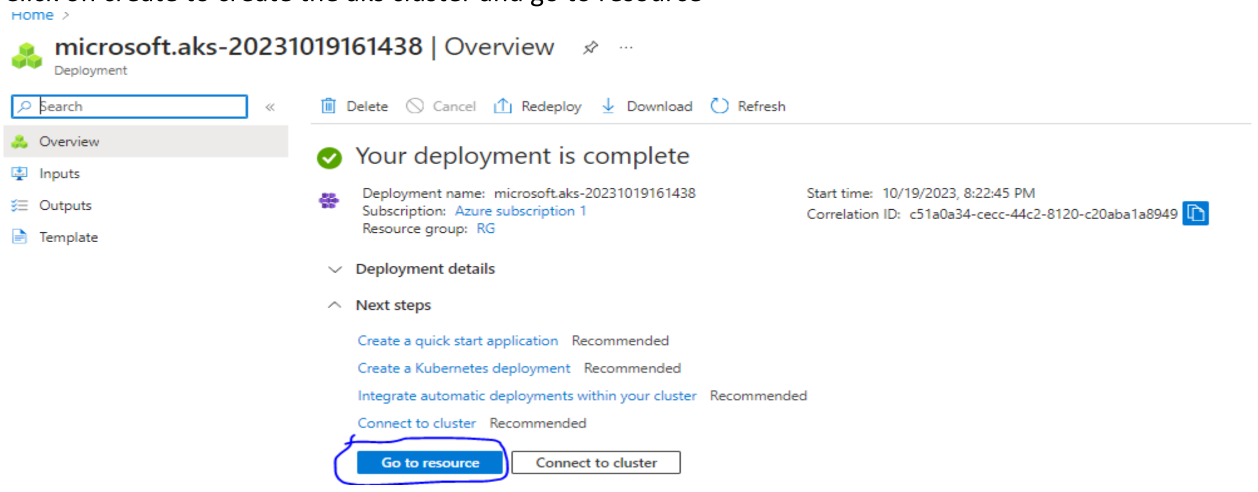
Node pools	1
Enable virtual nodes	Disabled

Access

Resource identity	System-assigned managed identity
Local accounts	Enabled

< Previous | Next > | Create | Download a template for automation

6) Click on create to create the aks cluster and go to resource



Home >

microsoft.aks-20231019161438 | Overview

Deployment

Search

Overview
Inputs
Outputs
Template

Delete | Cancel | Redeploy | Download | Refresh

Your deployment is complete

Deployment name: microsoft.aks-20231019161438
Subscription: Azure subscription 1
Resource group: RG

Start time: 10/19/2023, 8:22:45 PM
Correlation ID: c51a0a34-cecc-44c2-8120-c20aba1a8949

Deployment details

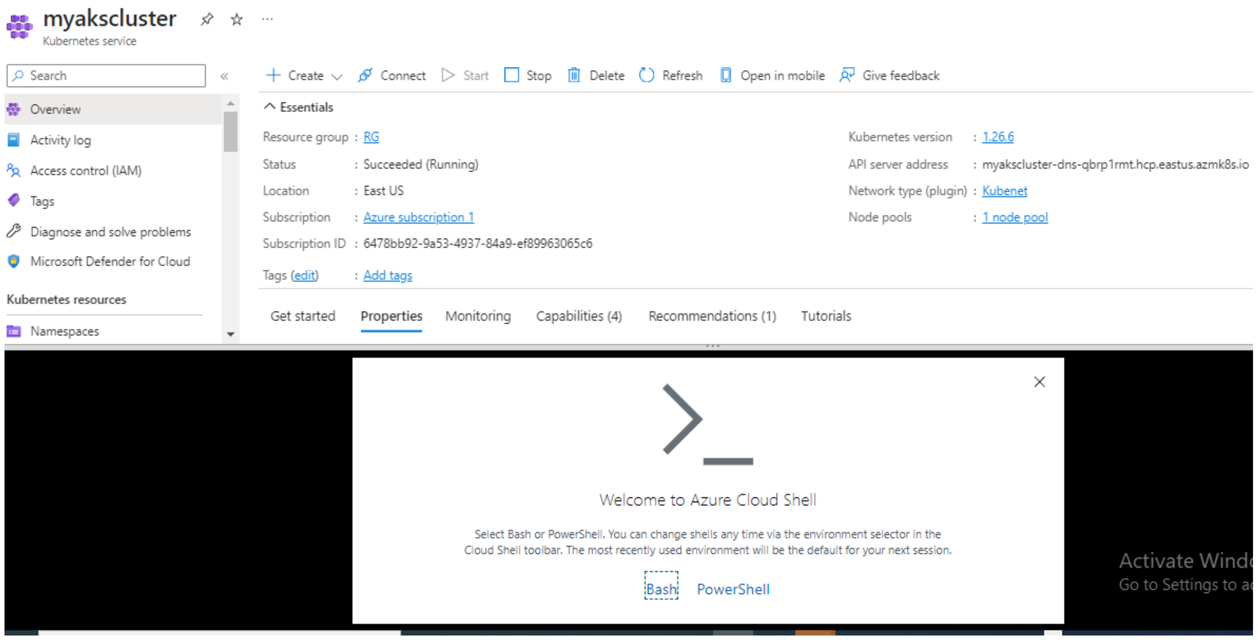
Next steps

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

Go to resource | Connect to cluster

We're going to deploy 2 container instances using a manifest file that contains the business logic

7) Click on the cloud shell and chose the Bash option and create a storage account if you hadn't created one before



Resource group : RG
Status : Succeeded (Running)
Location : East US
Subscription : Azure subscription 1
Subscription ID : 6478bb92-9a53-4937-84a9-ef89963065c6
Tags (edit) : Add tags

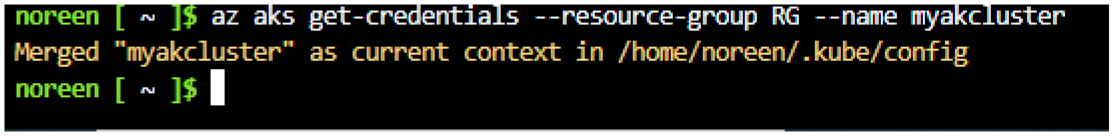
Kubernetes version : 1.26.6
API server address : myakscluster-dns-qbrp1rmt.hcp.eastus.azmk8s.io
Network type (plugin) : Kubenet
Node pools : 1 node pool

Welcome to Azure Cloud Shell

Select Bash or PowerShell. You can change shells any time via the environment selector in the Cloud Shell toolbar. The most recently used environment will be the default for your next session.

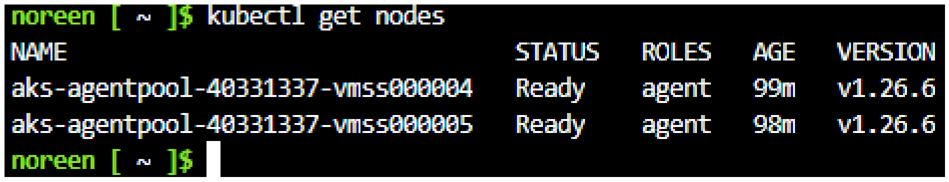
Bash PowerShell

8) Connect to the cloud shell and run the below command as shared in the screenshot



```
noreen [ ~ ]$ az aks get-credentials --resource-group RG --name myakcluster
Merged "myakcluster" as current context in /home/noreen/.kube/config
noreen [ ~ ]$
```

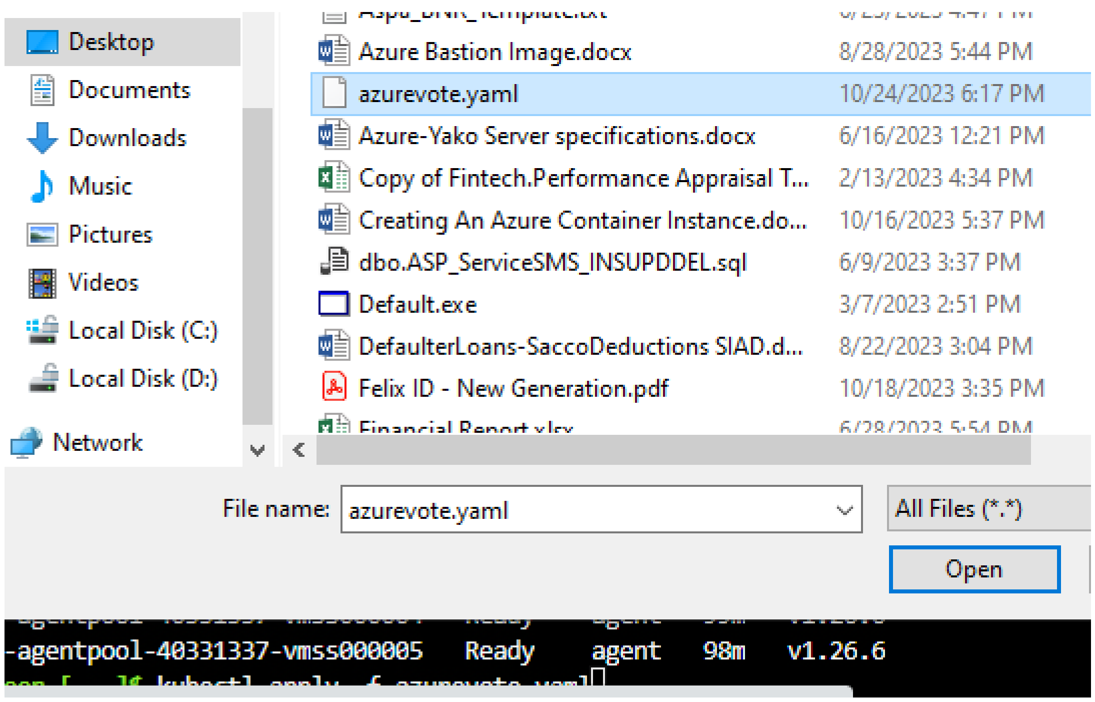
9) Next we want to get information on the nodes that are running. So we will execute the command below



```
noreen [ ~ ]$ kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-40331337-vmss000004   Ready    agent   99m   v1.26.6
aks-agentpool-40331337-vmss000005   Ready    agent   98m   v1.26.6
noreen [ ~ ]$
```

We're going to deploy the containers by uploading the yaml manifest file that I downloaded from this link https://github.com/Azure-Samples/azure-voting-app-redis/blob/master/azure-vote-all-in-one-redis.yaml.

10) Click on the upload icon on top of the bash shell to upload the *yaml* manifest file then use the apply command for the deployment as below;

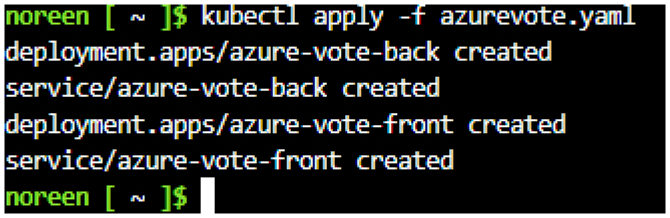


Azure Bastion Image.docx	8/28/2023 5:44 PM
azurevote.yaml	10/24/2023 6:17 PM
Azure-Yako Server specifications.docx	6/16/2023 12:21 PM
Copy of Fintech.Performance Appraisal T...	2/13/2023 4:34 PM
Creating An Azure Container Instance.do...	10/16/2023 5:37 PM
dbo.ASP_ServiceSMS_INSUPDDEL.sql	6/9/2023 3:37 PM
Default.exe	3/7/2023 2:51 PM
DefaulterLoans-SaccoDeductions SIAD.d...	8/22/2023 3:04 PM
Felix ID - New Generation.pdf	10/18/2023 3:35 PM

File name: azurevote.yaml

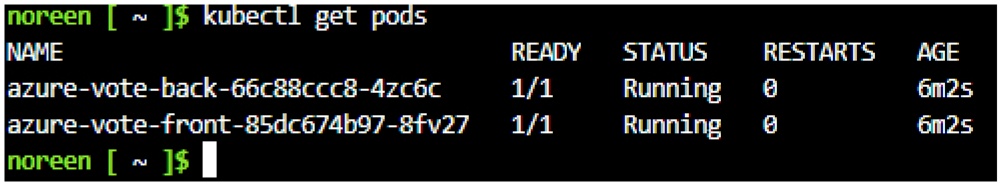
```
-agentpool-40331337-vmss000005   Ready    agent   98m   v1.26.6
```

11) Execute the "kubectl apply –f azurevote.yaml" command to see if our 2 containers have been deployed



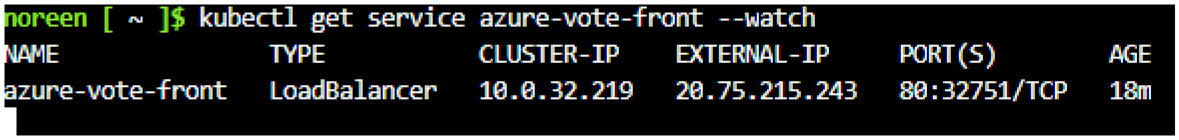
```
noreen [ ~ ]$ kubectl apply -f azurevote.yaml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
noreen [ ~ ]$
```

12) Inorder to see the containers, we're going to run the command "kubectl get pods"



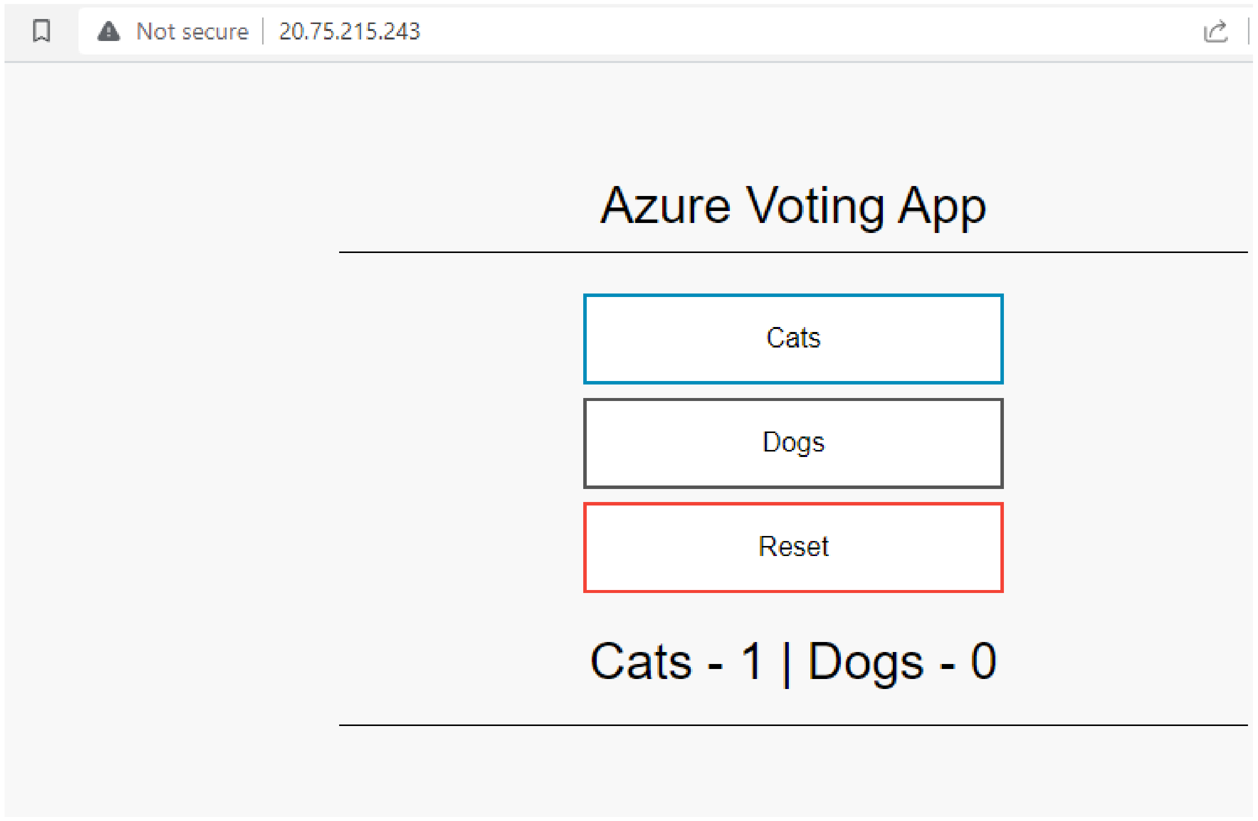
```
noreen [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
azure-vote-back-66c88ccc8-4zc6c     1/1     Running   0          6m2s
azure-vote-front-85dc674b97-8fv27   1/1     Running   0          6m2s
noreen [ ~ ]$
```

13) We will target our web application through the front end service i.e. your load balancer. Call the front end service by running the command "kubectl get service azure-vote-front –watch"



```
noreen [ ~ ]$ kubectl get service azure-vote-front --watch
NAME               TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)        AGE
azure-vote-front   LoadBalancer   10.0.32.219   20.75.215.243   80:32751/TCP   18m
```

14) Copy the external IP and hit on it to see your voting application below


Not secure
20.75.215.243
Azure Voting App
Cats
Dogs
Reset
Cats - 1 | Dogs - 0