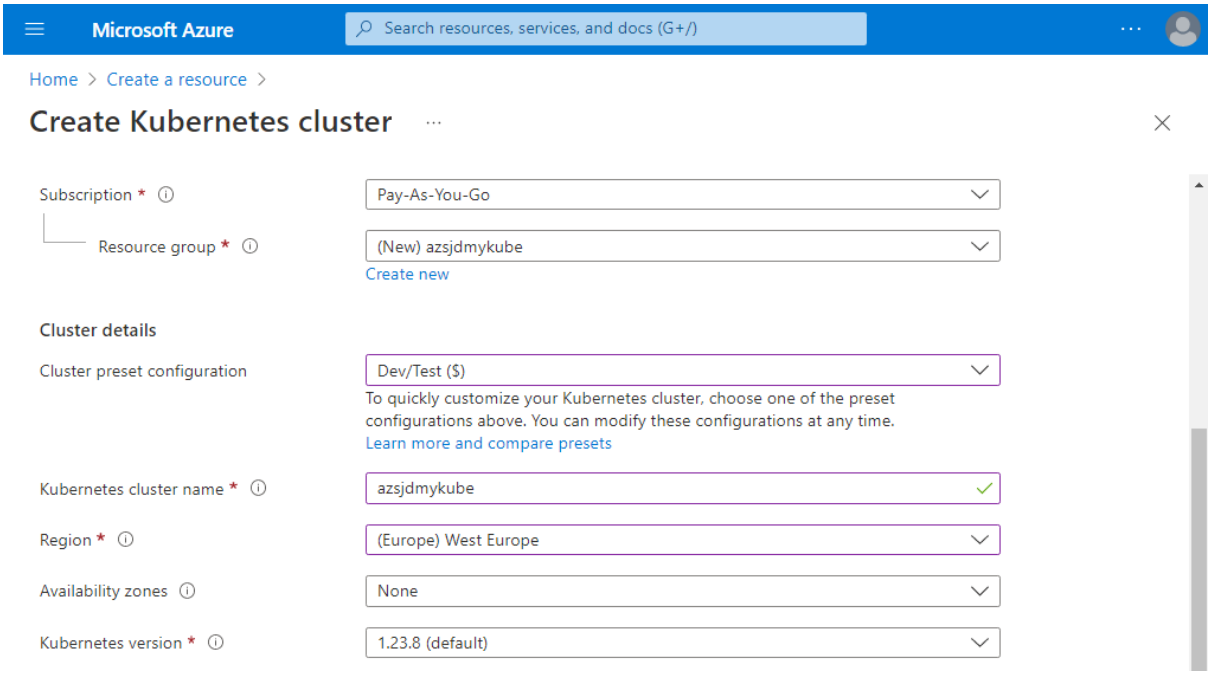
1. Create a Kubernetes Container

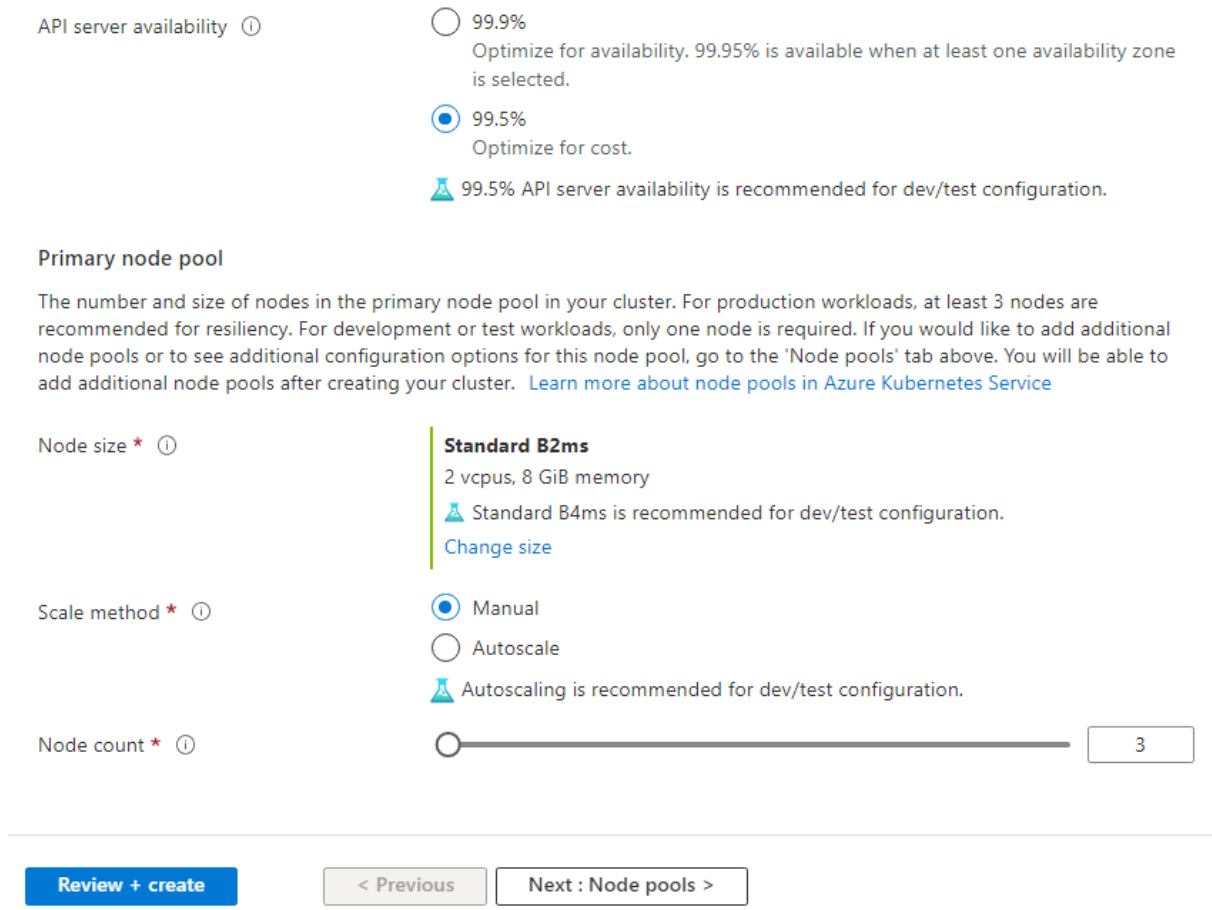
Create a brand new Azure Kubernetes Services (AKS) cluster using the portal, with the following characteristics:

- Dev/Test ($) cluster
- B2ms node size
- 3 nodes with manual scaling

1. From the Azure Home page, click on **create a resource**.
2. Find **Kubernetes Service** from the list.
3. Choose the correct subscription and resource group.
4. Choose the **Dev/Test ($)** cluster preset configuration.
5. Give the cluster a name.
6. Choose a region close to you.

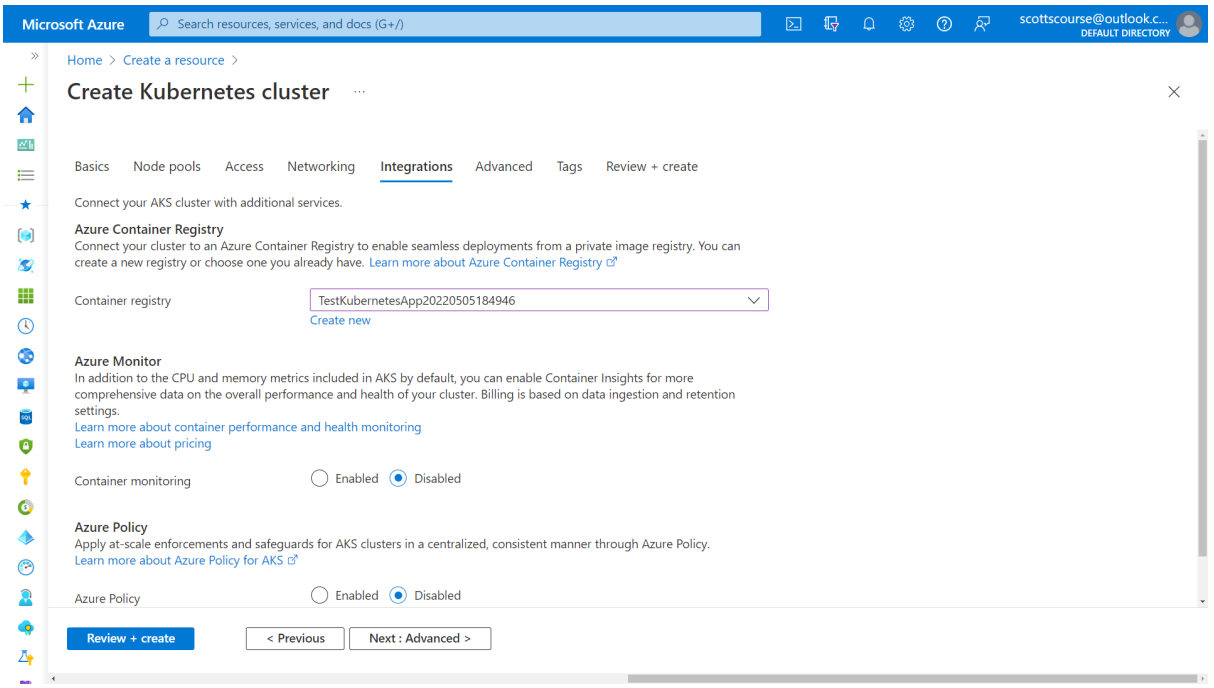


7. Choose a node size of **B2ms**. You will be creating **3 nodes with manual scaling**.

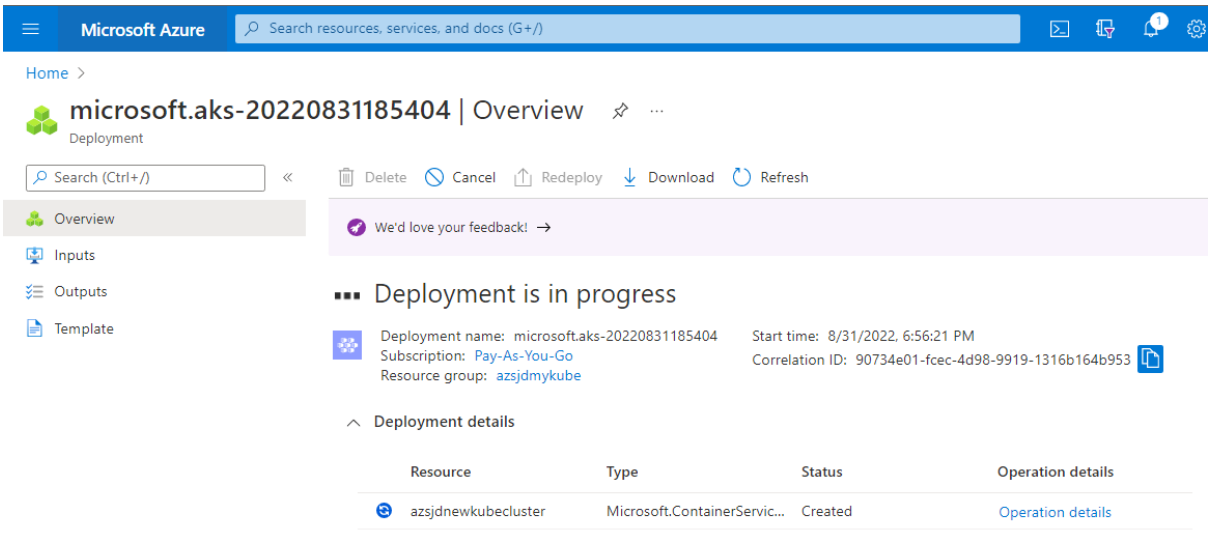


8. Keep the default for the rest on this page. Click **next**.
9. Keep the default for the node pools page. Click **next**.
10. Keep the default for the access page. Click **next**.
11. Keep the default for the networking page. Click **next**.

12. For integrations, choose your **Azure Container Registry** or create a **new** one. Click **Review and Create**.



13. Review this page, and click **Create**. ***NOTE:*** *Deploying this will take a while. Wait for it to complete.*



Resources

Documentation

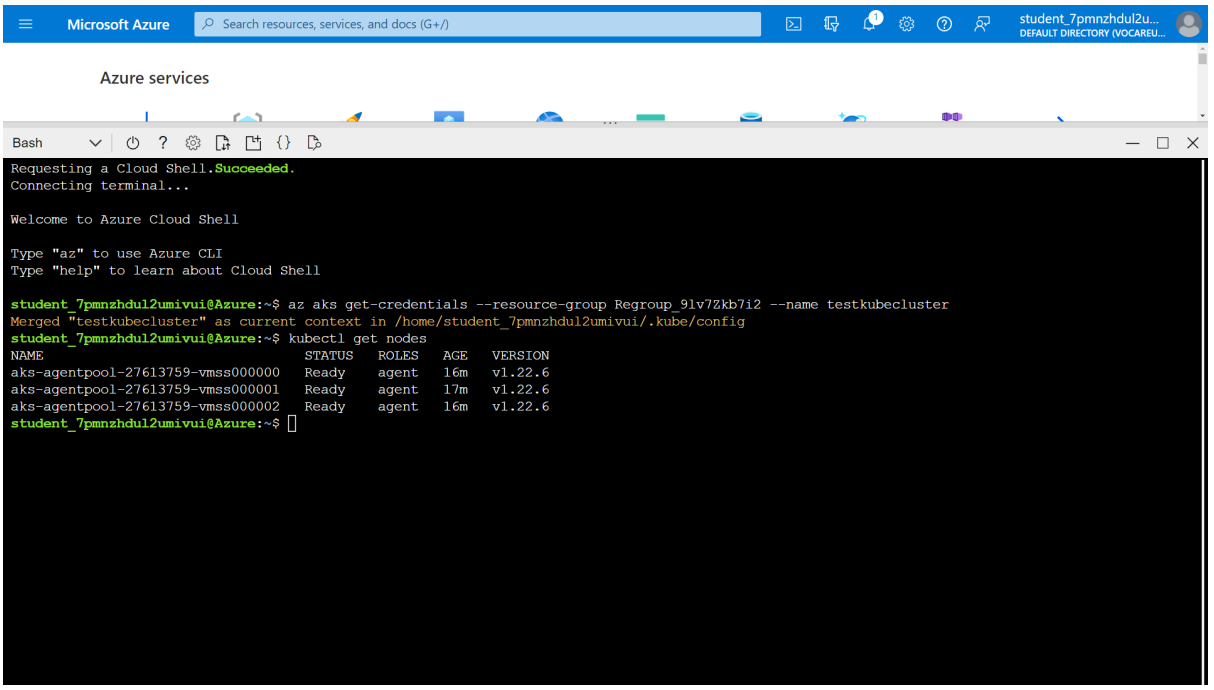
- Create AKS Cluster

2. Deploy the Container Image to the AKS Cluster

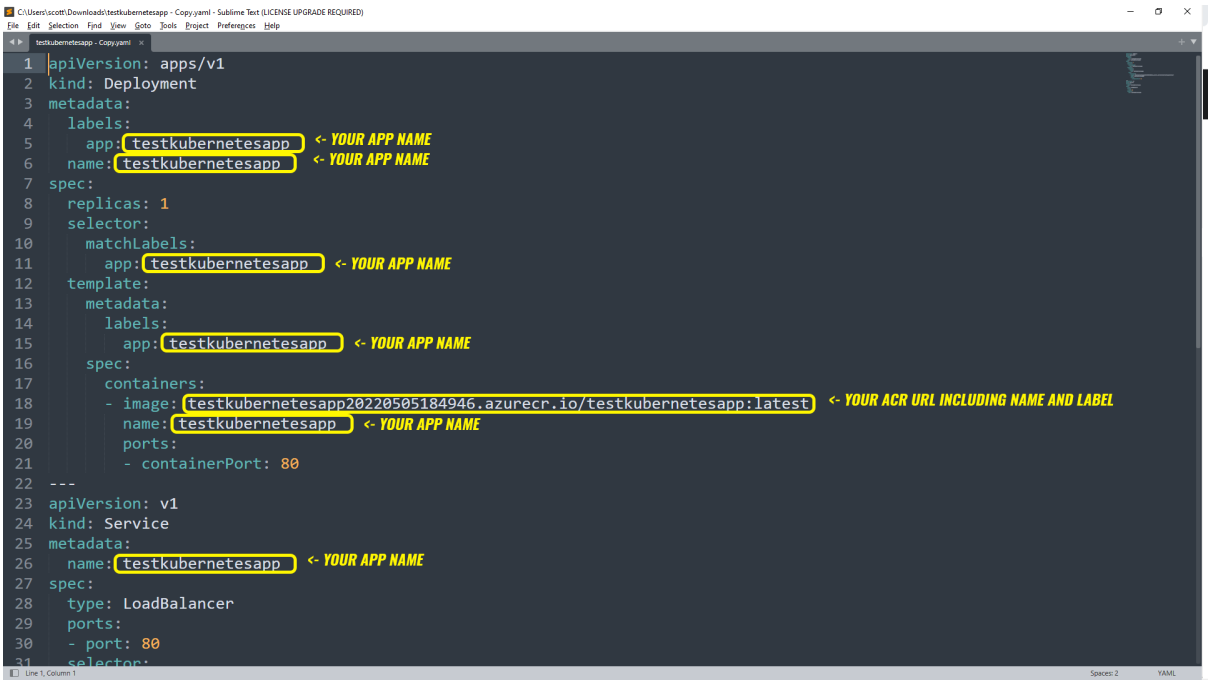
Deploy the image from ACR into the AKS cluster that you just created. You'll need to create a YAML file so that AKS knows how to deploy the image within the cluster.

1. Open the **Azure Cloud Shell**, and choose the **Bash/CLI** command prompt. Azure Cloud Shell is represented by the ">_" symbol at the top of the Azure Portal. Choose **Bash**.

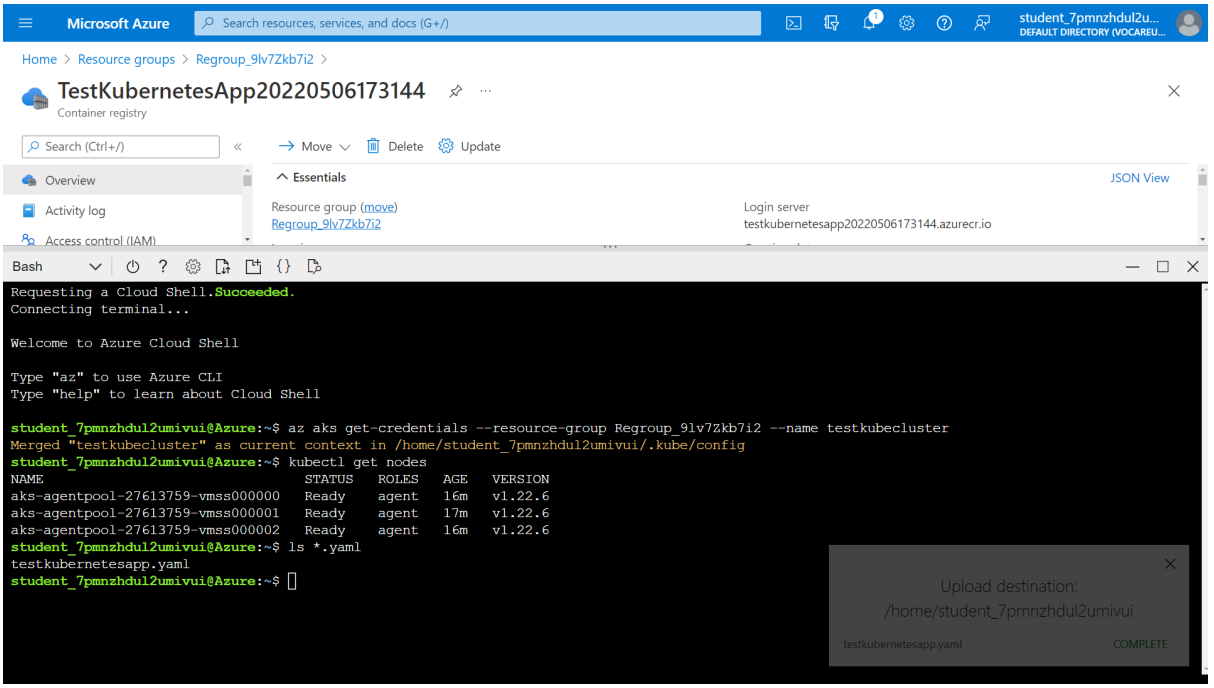
2. Get the credentials for AKS using the following command "**az aks get-credentials --resource-group <GROUP> --name <CLUSTER>**" where the **<GROUP>** is the name of your resource group and **<CLUSTER>** is the name of your AKS cluster.
3. Type "**kubectl get nodes**" to see the running nodes.



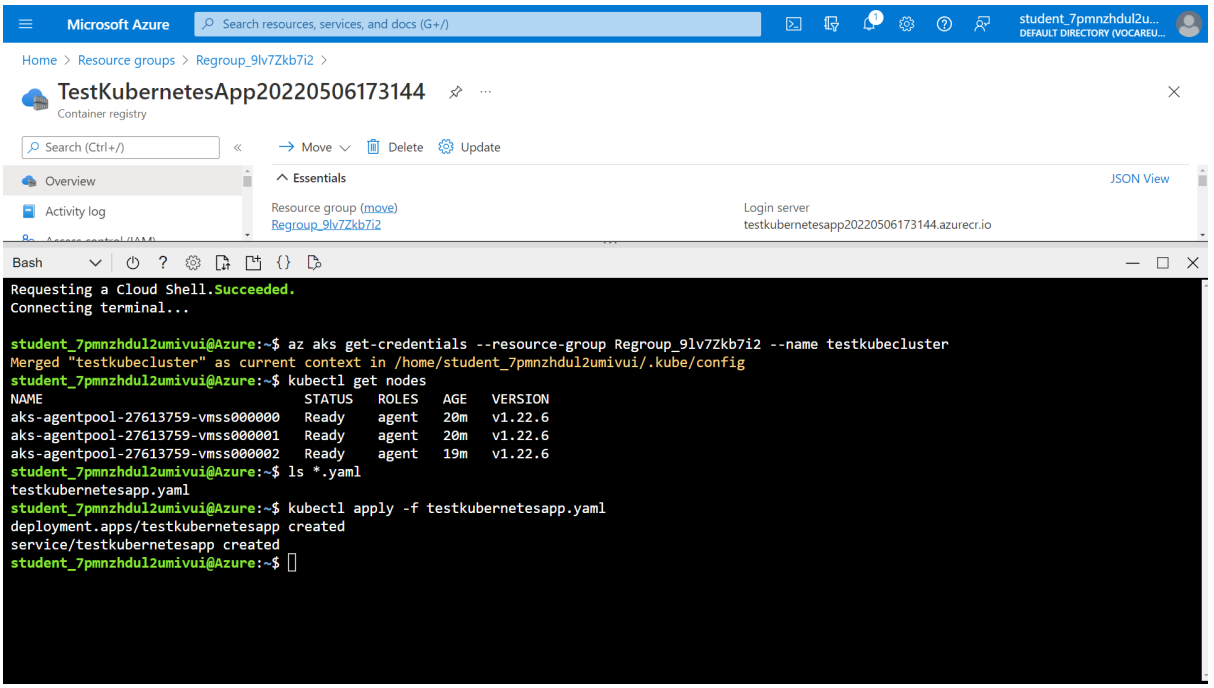
4. Download the sample YAML file called "testkubernetesapp.yaml".
5. Using a text editor, modify the YAML file to connect with your Azure Container Registry, replacing the **<youracr>** and **<yourappname>** with the appropriate values. Be sure and save the changes.



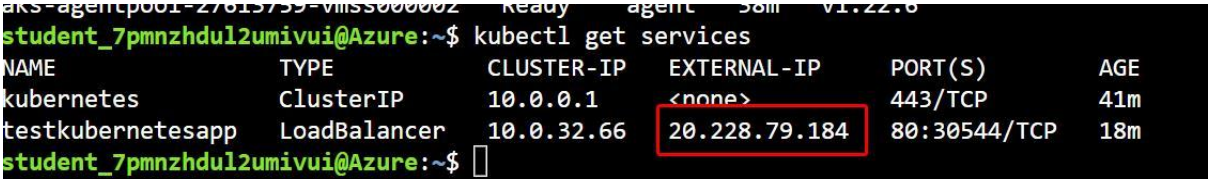
6. Upload the YAML file to Azure Cloud Shell using the upload function.
7. Type "**ls *.yaml**" to verify the upload worked.



8. Deploy the YAML file to Kubernetes using the command "**kubectl apply -f <yourfile>.yaml**" where <yourfile> is your YAML file name.

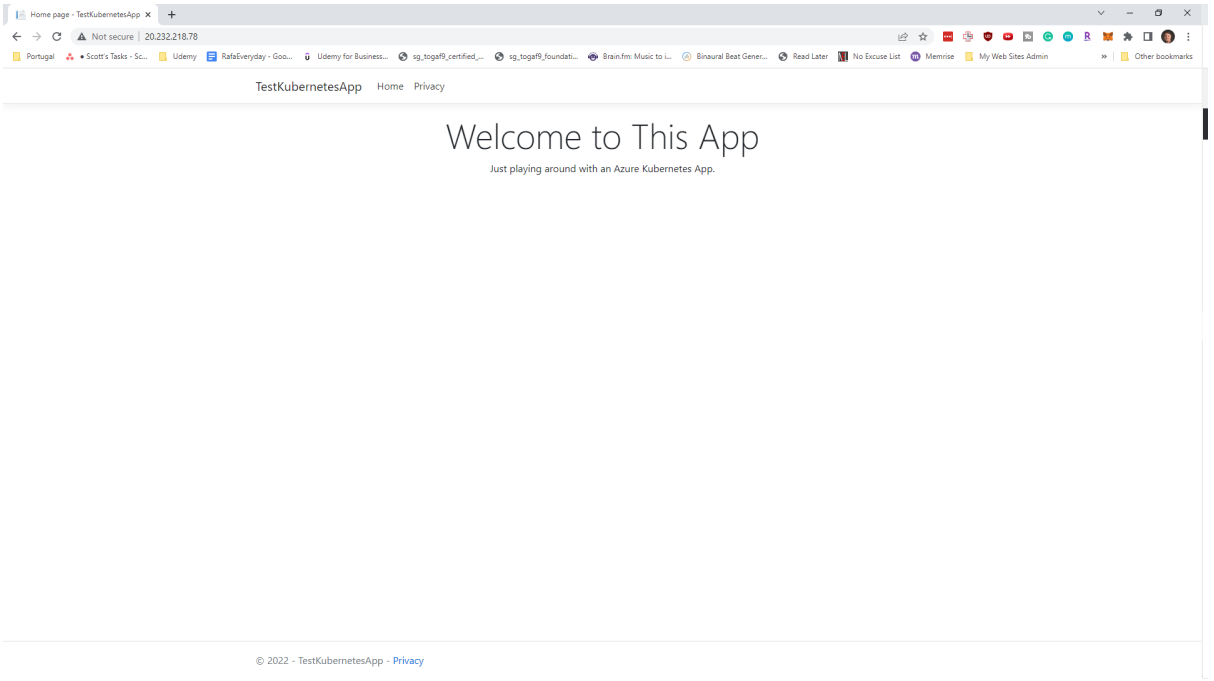


9. Type “**kubectl get nodes**” to see the nodes running.
10. Type “**kubectl get services**” to see if the load balancer is created. Wait until the load balancer is assigned an external IP address. You might have to repeat this step until an IP is created.



11. Make a note of the public IP address under “**external-IP**”. Copy the IP address to your clipboard.
12. Open a browser and go to that IP address. *Note: The application might take a few minutes to deploy, so wait for 2 minutes before loading the application.*

13. See that your application is successfully running in Azure Kubernetes Service.



Resources

Assets

- testkubernetesapp.yaml

Documentation

- Deploy to AKS

3. Scaling

You need to scale horizontally and vertically. Horizontal scaling means that the response to increased load is to deploy more Pods. This is different from vertical scaling, which for Kubernetes would mean assigning more resources (for example memory or CPU) to the Pods that are already running for the workload. Start with vertical scaling.

1. Open the Azure Cloud Shell, and choose the Bash/CLI command prompt. Azure Cloud Shell is represented by the ">_" symbol at the top of the Azure Portal. Choose Bash.

2. Type **kubectl scale --replicas=10 deployment/azure-vote-front** to increase the number of pods to 10.
3. Type **kubectl get pods -o wide** and check to see if you now have 10 pods running:

```
jordi_koenderink@Azure:~$ kubectl get pods -o wide
NAME                                READY   STATUS        RESTARTS   AGE    IP            NODE                                NOMINATED NODE   READINESS GATES
azure-vote-back-59cb7dc555-ttzr2    1/1     Running       0          13m    10.244.1.2    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-26wpv   1/1     Running       0          15s    10.244.0.11   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-28f9h   1/1     Running       0          15s    10.244.0.13   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-577nw   1/1     Running       0          13m    10.244.2.2    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-98c4m   1/1     Running       0          15s    10.244.2.4    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-99f5b   1/1     Running       0          4m13s  10.244.1.3    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-bb7rw   1/1     Running       0          15s    10.244.0.14   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-dqkb4   1/1     Running       0          15s    10.244.1.6    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-dv8tk   1/1     Terminating   0          15s    10.244.1.5    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-h7d86   1/1     Terminating   0          15s    10.244.1.4    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-kgdb8   1/1     Terminating   0          15s    10.244.2.5    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-lmz4b   1/1     Terminating   0          15s    10.244.2.3    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-w5st2   1/1     Running       0          15s    10.244.2.6    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-wcq92   1/1     Terminating   0          15s    10.244.1.7    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-xjt7m   1/1     Running       0          15s    10.244.0.12   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-zn9nw   1/1     Running       0          4m13s  10.244.0.10   aks-agentpool-21567423-vmss000002   <none>           <none>
```

4. Perform the same commands but now scale down to 1 pod by typing **kubectl scale --replicas=1 deployment/azure-vote-front**
5. Double check how many pods you now have running by using the command **kubectl get pods -o wide.**

   Note: it can take some time to scale down, hence the Terminating status:

```
jordi_koenderink@Azure:~$ kubectl get pods -o wide
NAME                                READY   STATUS        RESTARTS   AGE    IP            NODE                                NOMINATED NODE   READINESS GATES
azure-vote-back-59cb7dc555-ttzr2    1/1     Running       0          15m    10.244.1.2    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-26wpv   1/1     Terminating   0          2m29s  10.244.0.11   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-28f9h   1/1     Terminating   0          2m29s  10.244.0.13   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-577nw   1/1     Terminating   0          15m    10.244.2.2    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-98c4m   1/1     Terminating   0          2m29s  10.244.2.4    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-99f5b   1/1     Running       0          6m27s  10.244.1.3    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-bb7rw   1/1     Terminating   0          2m29s  10.244.0.14   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-dqkb4   1/1     Terminating   0          2m29s  10.244.1.6    aks-agentpool-21567423-vmss000000   <none>           <none>
azure-vote-front-5f4d7db9c8-w5st2   1/1     Terminating   0          2m29s  10.244.2.6    aks-agentpool-21567423-vmss000001   <none>           <none>
azure-vote-front-5f4d7db9c8-xjt7m   1/1     Terminating   0          2m29s  10.244.0.12   aks-agentpool-21567423-vmss000002   <none>           <none>
azure-vote-front-5f4d7db9c8-zn9nw   1/1     Terminating   0          6m27s  10.244.0.10   aks-agentpool-21567423-vmss000002   <none>           <none>
```

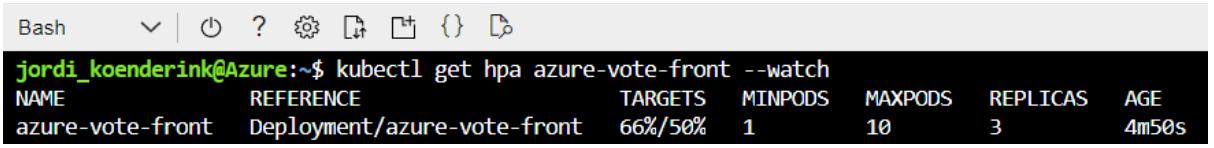
6. Now move on with horizontal scaling. Create the HorizontalPodAutoscaler by typing the command **kubectl autoscale deployment azure-vote-back --cpu-percent=50 --min=1 --max=10**
7. Check the status by using the command **kubectl get hpa.** The current CPU consumption is 2%:

```
jordi_koenderink@Azure:~$ kubectl get hpa
NAME               REFERENCE                     TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
azure-vote-back    Deployment/azure-vote-back    2%/50%    1         10        1          43s
```

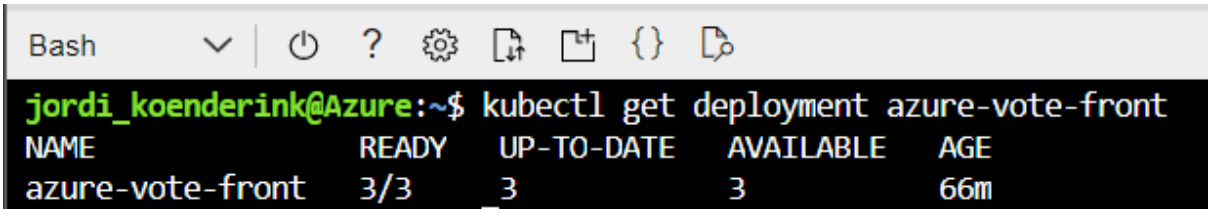
8. Type **kubectl get service** and write down the external IP.
9. Increase the load to test the autoscaler by typing **kubectl run -i --tty load-generator --rm --image=busybox:1.28 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- 20.23.34.186; done".** Replace the IP address with the IP address from step 8.

   Note: type **CTRL+C** to end the loop.

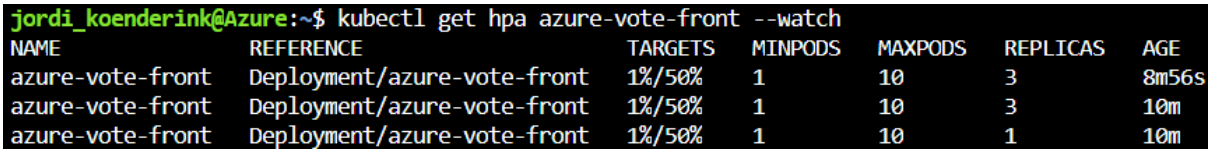
10. Now, run **kubectl get hpa azure-vote-back --watch**
11. After a couple of minutes you should see a higher CPU load, and more replicas:



12. Type **CTRL+C** to end the Watch.
13. Type **kubectl get deployment azure-vote-front** to see the replica count matching the figure from the HorizontalPodAutoscaler in another view:



14. After a couple of minutes the number of replicas have scaled down:



© 2023 Scott J Duffy, Jordi Koenderink, and SoftwareArchitect.ca, all rights reserved