Azure section 14.02 - Container and AKS hands on by deploying the cart module.

Now we have seen how to deploy the image on the ACR. We did it and the image is there in the repository.

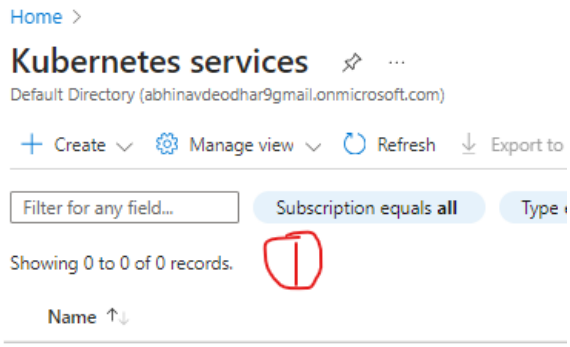
But ACR just is a registry which manages the images and builds them.

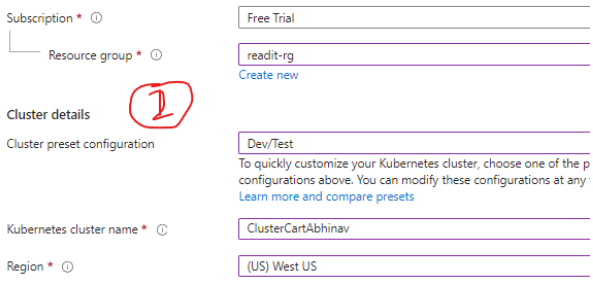
We now need a service which can also manage the containers.

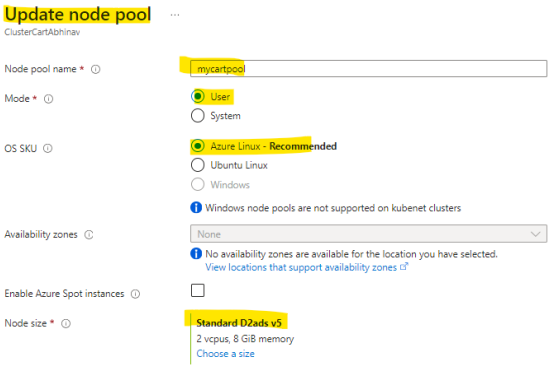
For this we will be building Kubernetes Cluster.

Step 1:

1. Go to the search section and then search for the Kubernetes. Select the service
2. Go to the create and select the cluster service.
3. Choose the Resource group and set the configuration to DEV/Test.
4. Give the name to the cluster choose the region and version of the cluster.
5. There is no availability zone. We need to set the node size to D series.
6. Set the scale method to manual and set the node count to three.







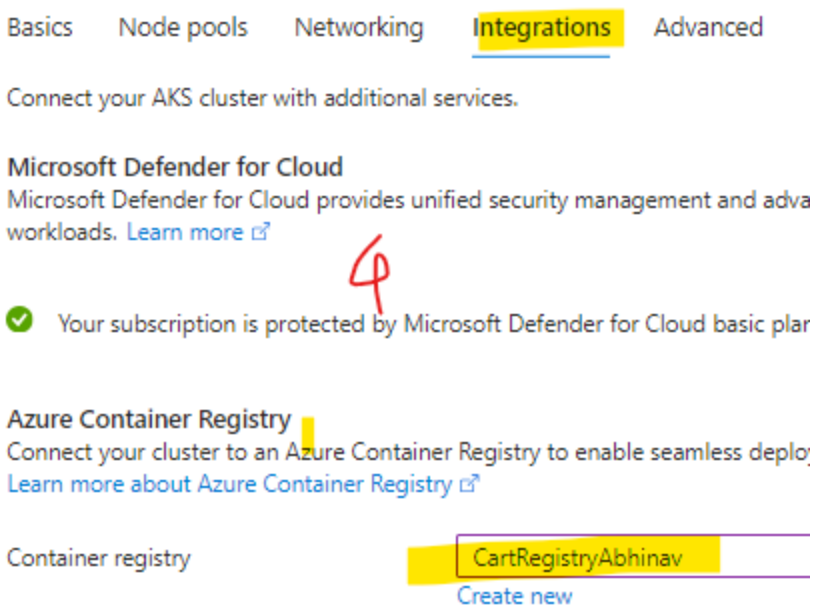
Step 1.01:

1. There will be the error of the sufficient quota. This means that the region which we have selected is already using all the allowed resources which come under the free subscription.
2. So to solve this error go to the subscription and then click on the usage+quota option under the navigation menu. We can see the what are the resource which we are using at maximum value.
3. So we need to change the region simply to tackle this error. i choose west us2.

Step 2:

1. Above the page we can see the integration tab we need to click on it.
2. Under that we need to choose the container registry and select the container registry that we have just created. Review and create.
3. Now AKS is created.

So far what we have is We have ACR which is having our docker image in its repository.
We then needed a service which could manage the container formed by running our docker image.
Then we created the Kubernetes Cluster for that purpose.
Now this AKS is where we will deploy our running image i.e our container.



Step 3:

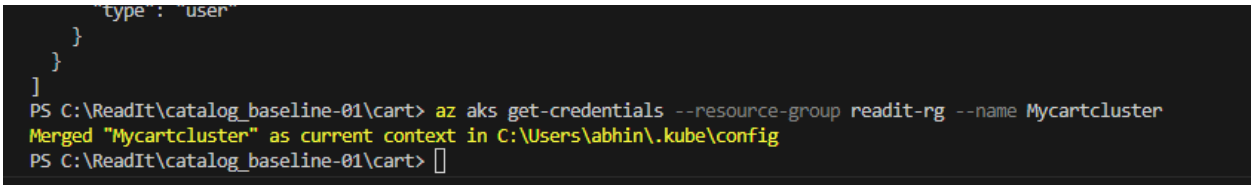
1. Now we need to install AKS CLI to run the Kubernetes commands.
2. Use the following command to do so az aks install-cli
3. Then we need to set the AKS CLI path in environment variable so that we can use it from anywhere.

For that the command is
set PATH=%PATH%; “Path of the AKS CLI”

Step 4:

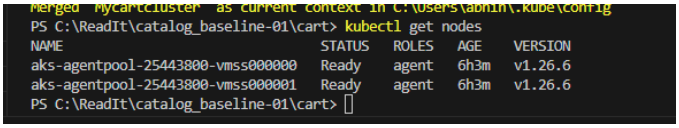
1. Now login to our azure account through azure CLI.
2. Then we need to connect to the AKS cluster for that we first need to get the credentials.

> az aks get-credentials —resource-group <name_of_rg> —name <name_of_AKS>



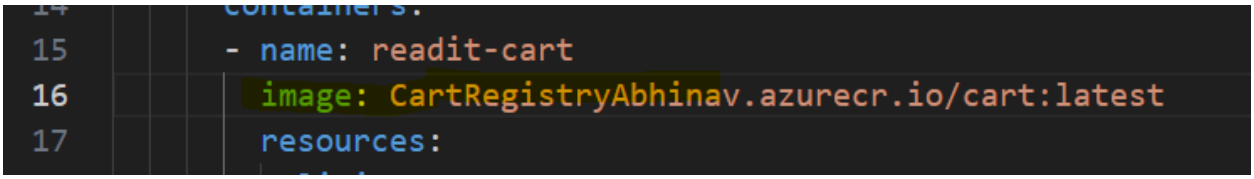
Step 5:

1. We will connect to the kubernetes. for that we need to run this command **kube ctl get nodes.**
2. This command will show us the node list which are in the cluster which we have created.
3. The status of the node in the result should be ready.



Step 6:

1. So there is the deployment.yaml file which contains the details of how the code must be deployed to the AKS.
2. We need to specify the name of the ACR in the code.
3.



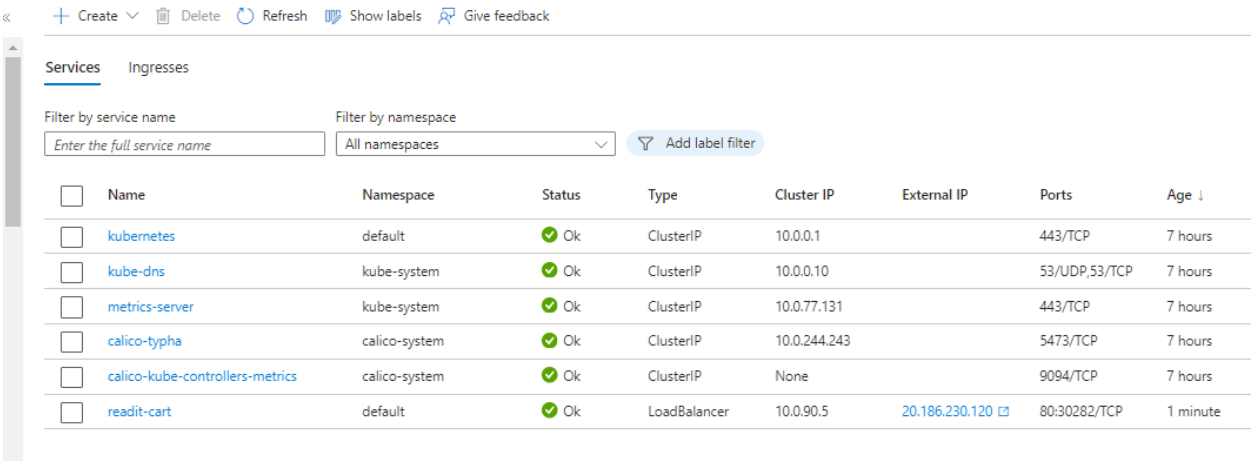
Step 7:

1. Now we are ready to deploy.
2. Use the command kube ctl apply -f deployment.yaml
3. AKS will first read the file and then deploy the container.
4.

```
PS C:\ReadIt\catalog_baseline-01\cart> kubectl apply -f deployment.yaml
deployment.apps/readit-cart created
service/readit-cart created
PS C:\ReadIt\catalog_baseline-01\cart>
```

Step 8:

1. Now we can cross check if the container is up and running or not.
2. Go to the AKS services and ingresses.
3. There we can see our service. There is an IP address provided to the port associated to the service. That port contains the container which we have pushed to ACR.



SO YOU CAN SEE THE EXTERNAL ip LINK AT THE CONTAINER NAME CLICK ON THAT AND THE CART MODULE WILL RUN.

Here we are done with the AKS container deployment.