**Step 1: Azure Container Registry (ACR): Create Azure Container Registry to store Docker images.**

az acr create --resource-group <resource-group> --name <acr-name> --sku Basic

**Step 2: Azure Kubernetes Service (AKS): Create AKS cluster.**

az aks create --resource-group <resource-group> --name <aks-cluster-name> --node-count 1 --enable-addons monitoring --generate-ssh-keys

**Step 3: Azure Kubernetes Config (kubectl): Configure kubectl to use the AKS cluster.**

az aks get-credentials --resource-group <resource-group> --name <aks-cluster-name>

**step 4: Build Docker Images: Build and push Docker images to ACR.**

|  |
| --- |
| **docker build -t <acr-name>.azurecr.io/product-service:v1 .**  **docker push <acr-name>.azurecr.io/product-service:v1**  **docker build -t <acr-name>.azurecr.io/order-service:v1 .**  **docker push <acr-name>.azurecr.io/order-service:v1** |

**Step 5: Kubernetes Deployment**

Created Kubernetes Services for both microservices. [Please refer product-service-service.yaml and order-service-service.yaml files in github]

Next part deployement:

Kubernetes Deployments: Created Deployments for blue-green deployment. Please refer to product-service-deployment.yaml ,order-service-deployment.yaml files in github

**Step 6: Blue-Green Deployment**

|  |
| --- |
| **kubectl apply -f product-service-deployment.yaml**  **kubectl apply -f order-service-deployment.yaml**  **kubectl apply -f product-service-service.yaml**  **kubectl apply -f order-service-service.yaml** |

Use updated deployment files with new images (v2).

Update the image in product-service-deployment.yaml and order-service-deployment.yaml to use version v2.

Apply the changes using kubectl apply -f <deployment-file>.

**Validation**: Get the external IP for the product-service and validate.

kubectl get services product-service

**Rollback** Deployments: In case of issues, rollback to the previous version.

kubectl rollout undo deployment/product-service

kubectl rollout undo deployment/order-service