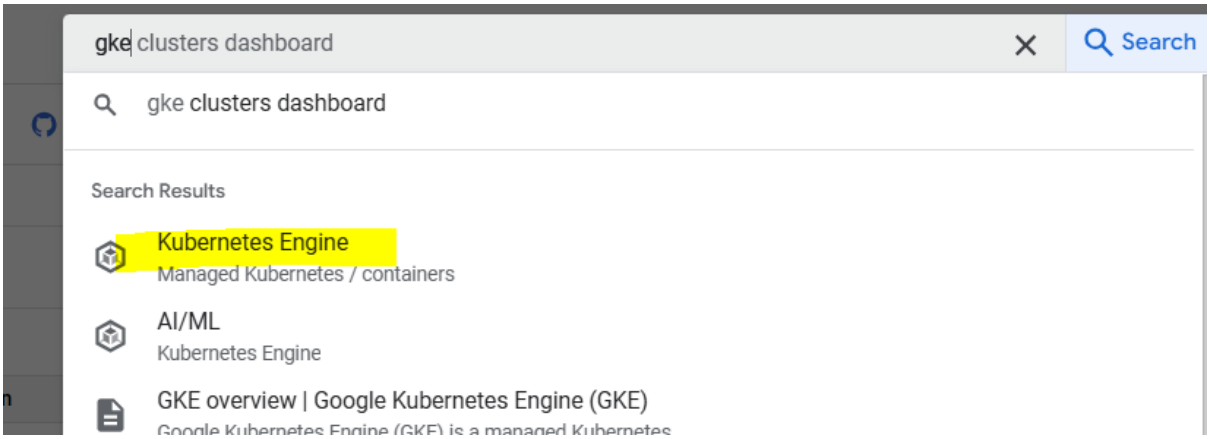
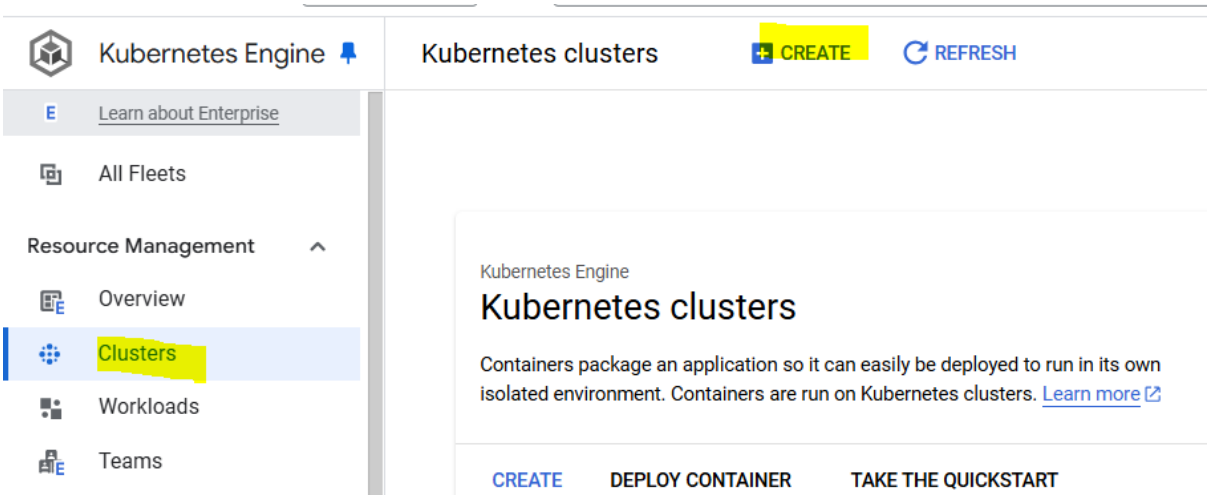
In this lab, we are going to deploy the cart container that we deployed earlier to Cloud Run to GKE in autopilot mode.

1) Go to GCP and search for GKE and click Kubernetes Engine

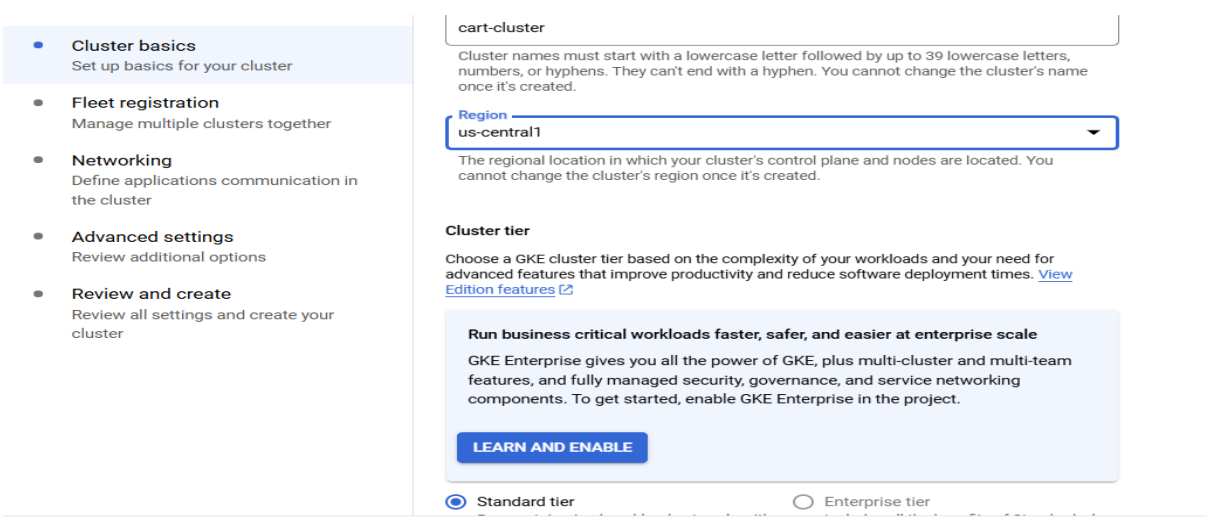


2) Enable the API if asked

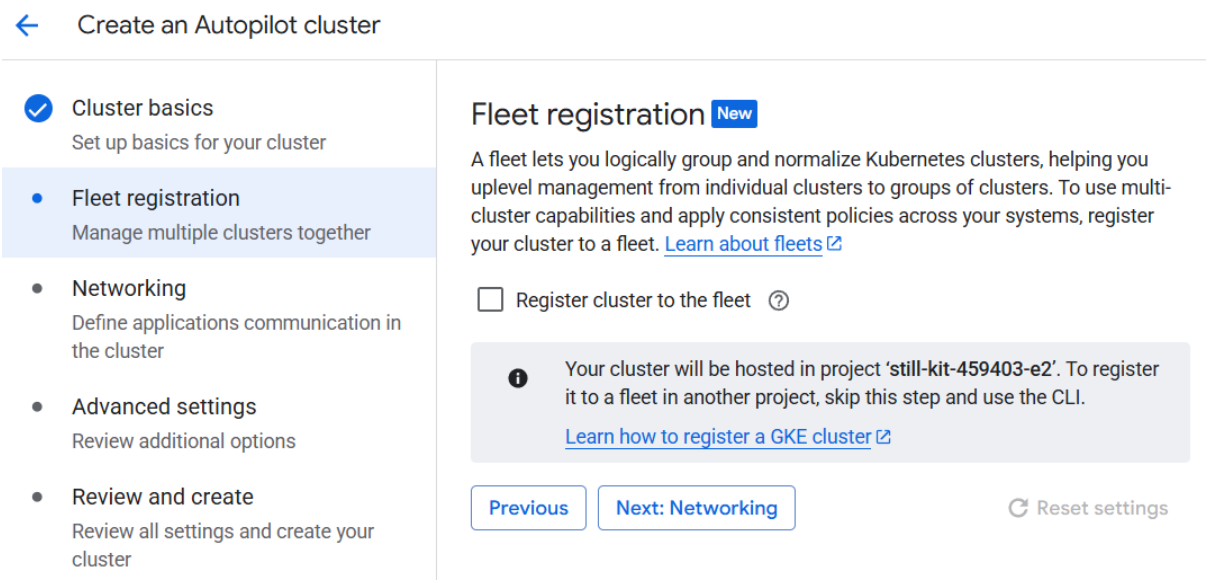
3) Go to clusters and click create



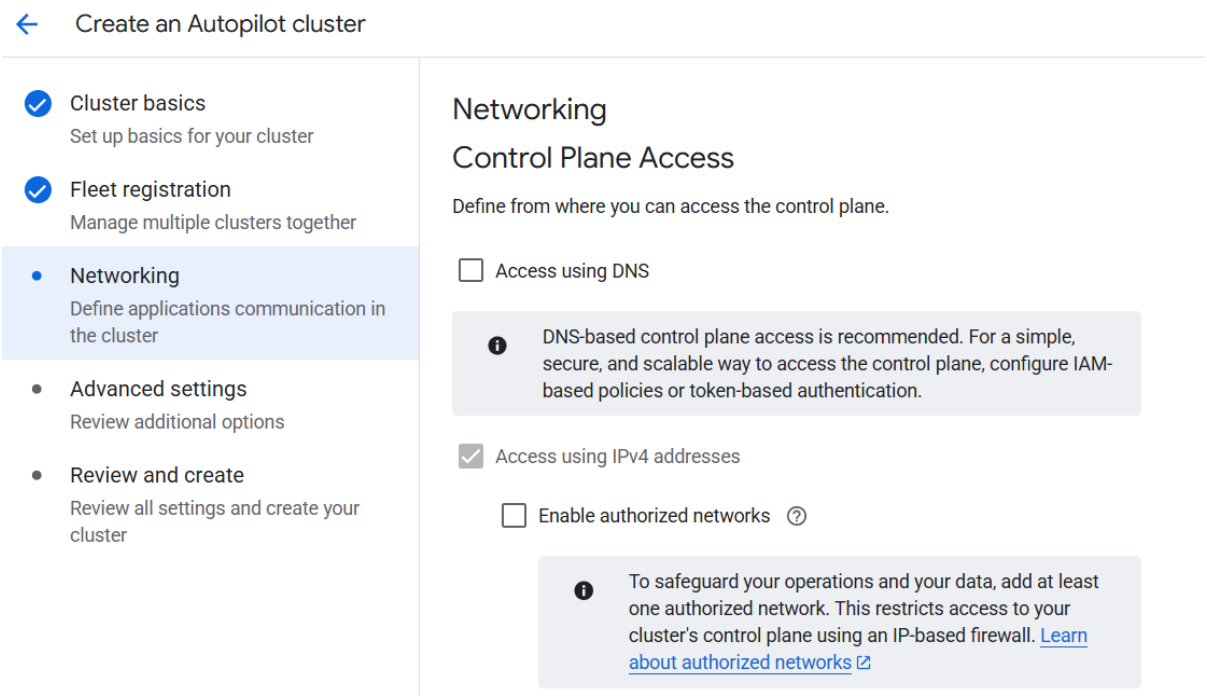
4) Give cluster a name, your region and keep as Standard tier



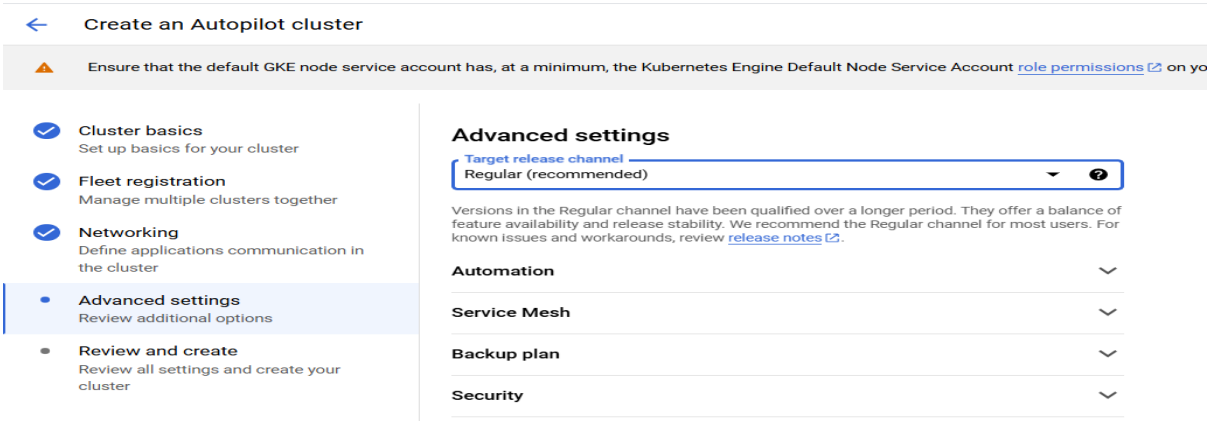
5) Go to the next tab, fleet registration, keep all default



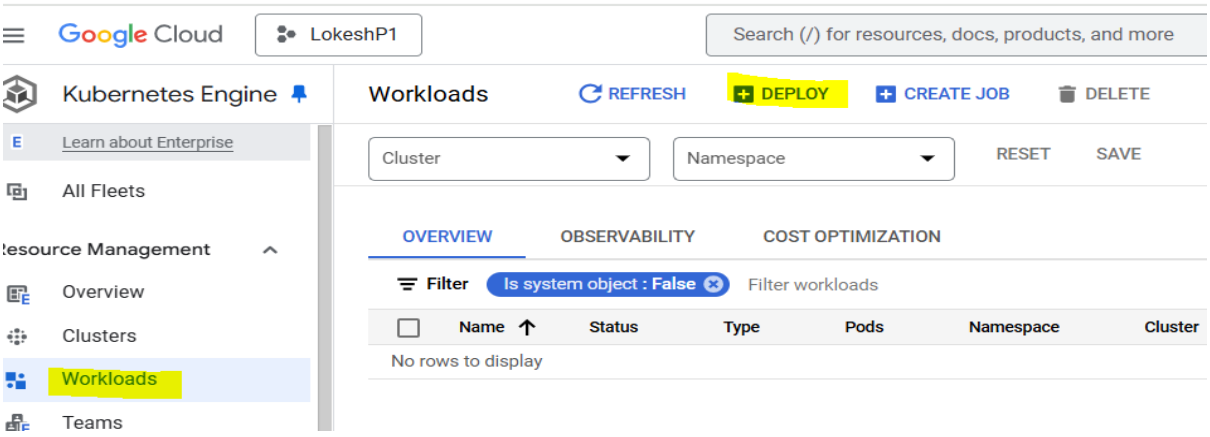
6) Go to next tab, Networking, check VPC details (keep default)



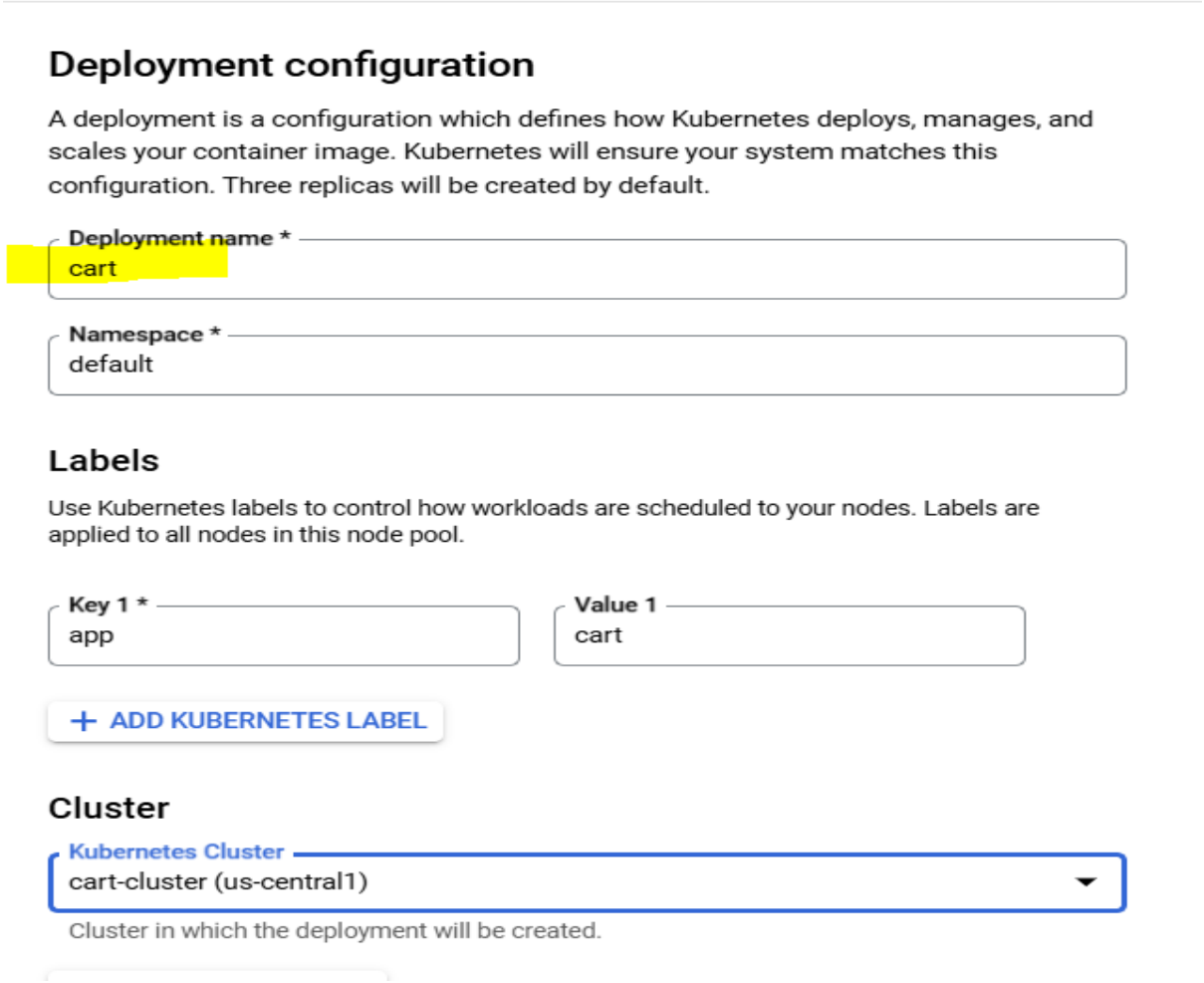
7) Go to the Advanced settings tab, in this list we can see three options, Rapid, Regular, and Stable, so basically what we set here is what is the Kubernetes version that we are going to use and how upgrades are going to be managed.
8) This means that when a new version of Kubernetes is released and has passed Google's validation, our cluster will be upgraded to support it.
9) Rapid channel means that we are going to be upgraded as fast as possible to the new version, and stable channel is the slowest channel. And only after we are really sure that we want to upgrade our cluster to the new version of Kubernetes, then the upgrade takes place.
10) So, we will stick to the regular channel, keep all default and click create



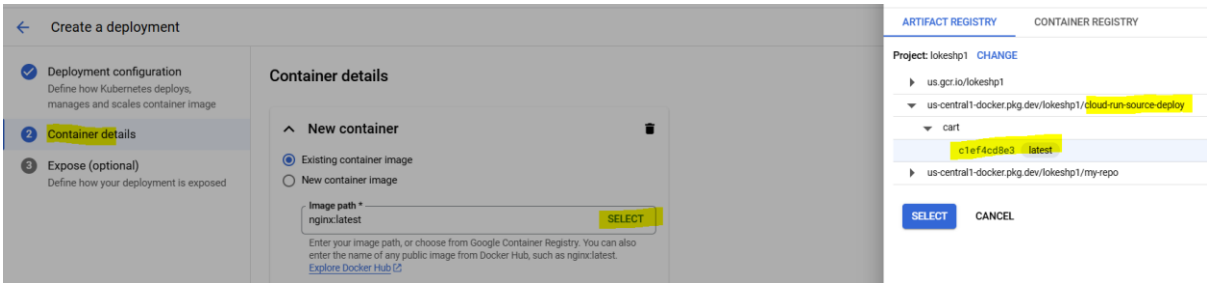
11) Once the cluster is created, which will take a few minutes, we will deploy our cart image to this cluster. For this, go to Workloads and click Create Deployment



12) Give this deployment a name like cart, and make sure the cluster which we just created is selected under Cluster option



13) Select latest image from repo we created earlier via code

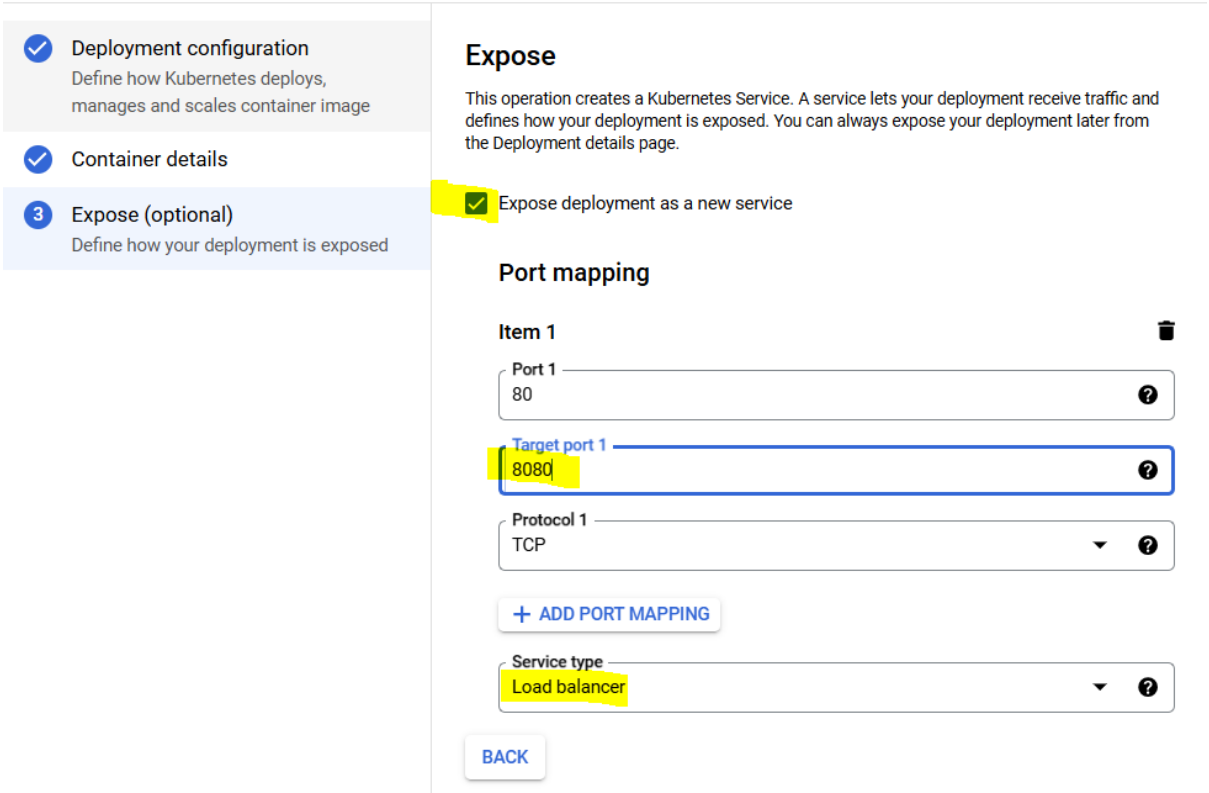


14) In next option, check box to expose the deployment. By this we created a new public endpoint that allows us to access pods in the cluster.

15) Here, service will be exposed to port 80, which is by default and the **target port** is the port at which the pod.

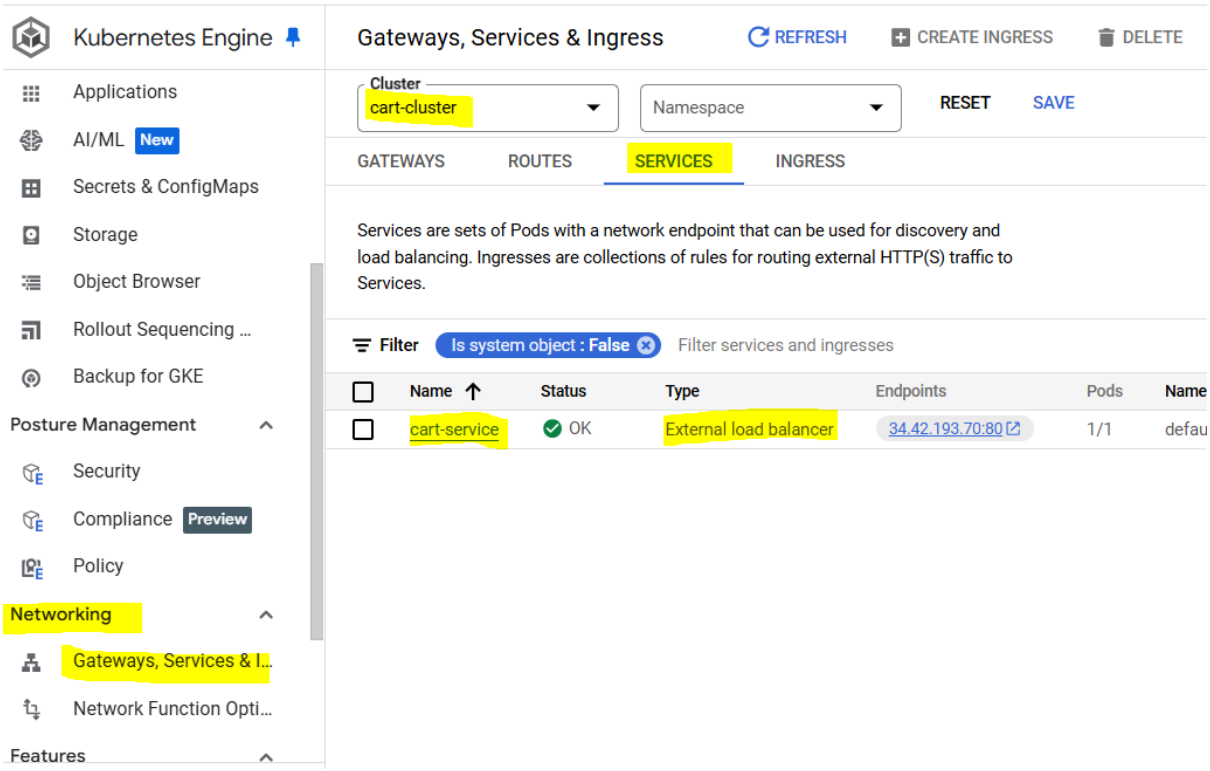
16) In **service type**, we have three options, first is cluster IP, by which only resources in the cloud can access this resource, second is Node port, which creates a service for each node, and third is load balancer, which creates a load balancer with external IP.

17) So here, we will select **load balancer** option (Note, in case you see an error like does not have minimum availability,, ignore that and it will go away in few minutes)

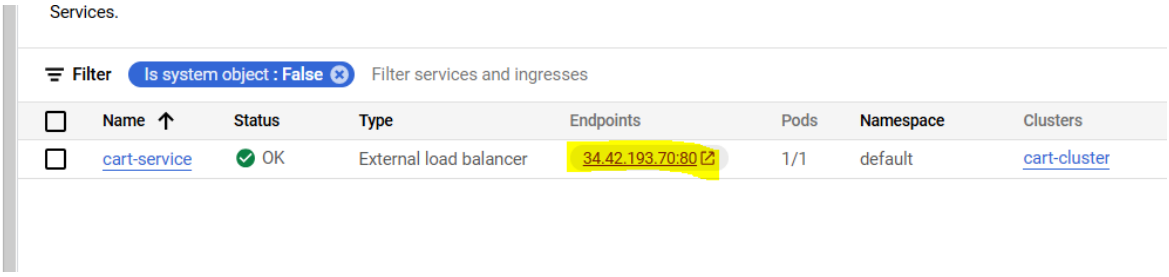


18) Once deployment is completed, scroll down left pane and go to Gateways, services, and Ingress option under networking.

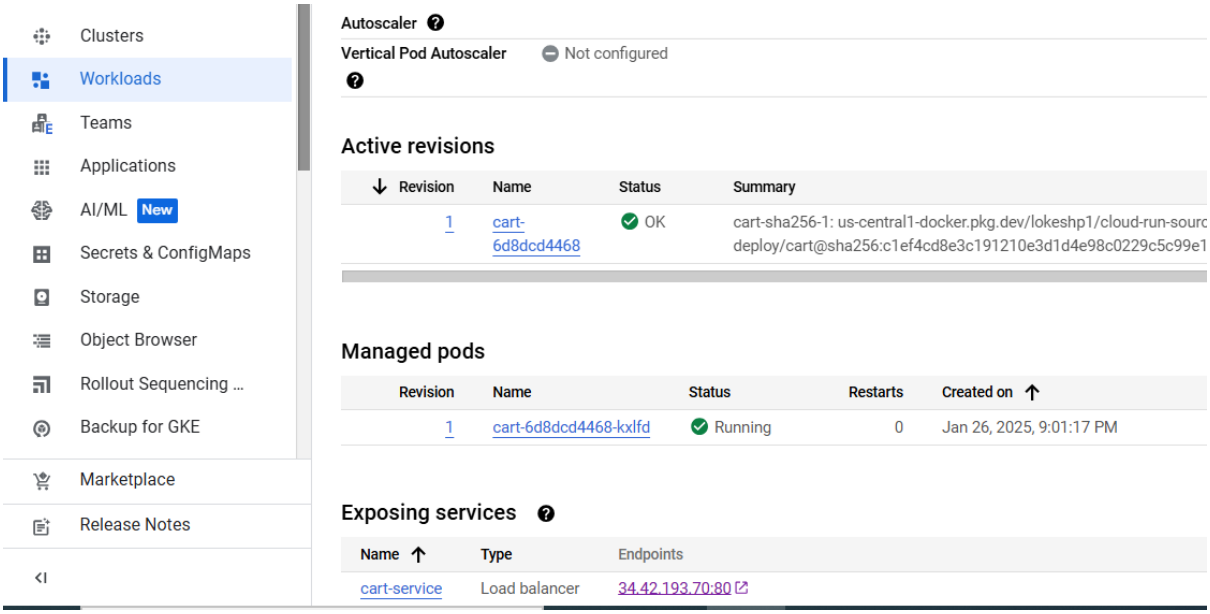
19) Here select your cluster, go to services and here it will show our cart service and type as external load balancer, which we selected while creating



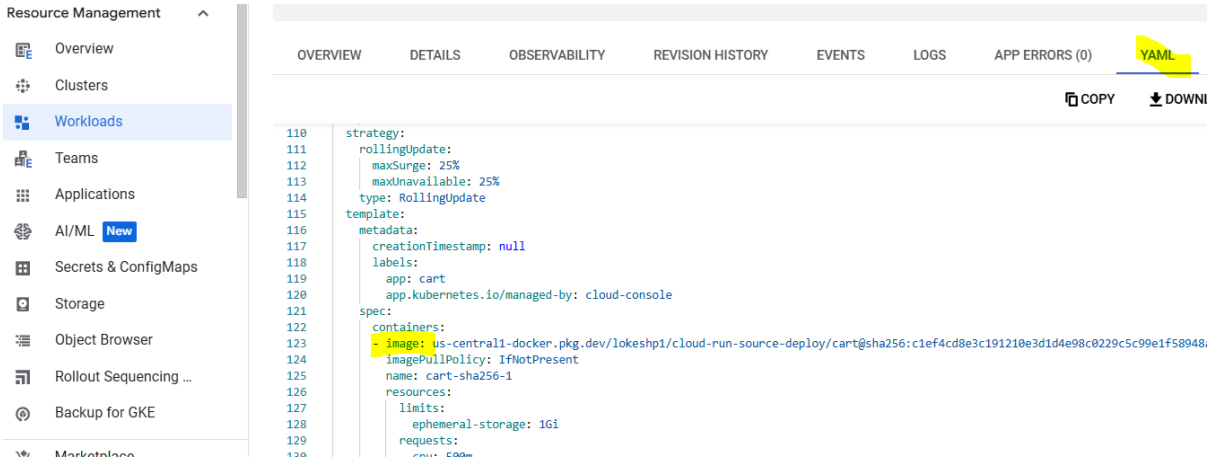
20) Click on the IP address, and this should open our cart webpage as per selected image



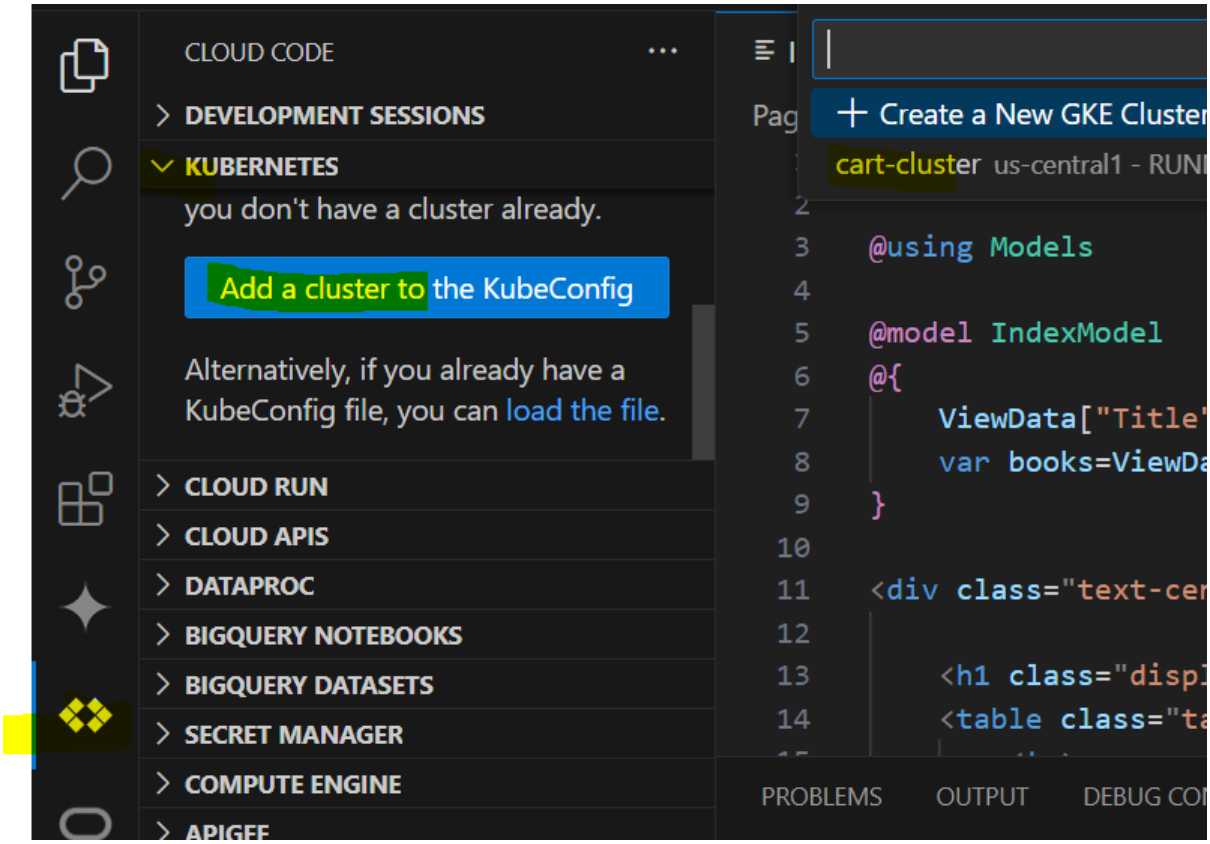
21) Now go back to workloads tab again, open the cart deployment and scroll down, here we can see the pods, revision, exposed service details and all



22) Now go to YAML option, and here we can see all the configurations related to this GKE deployment and if you scroll down, there we can see image details as well which is one of most important part



23) Now go to vscode, gcp cloud option, expand Kubernetes and Add a cluster option. Select google cloud and then select cart-cluster



24) This will add our Kubernetes cluster to the vscloud and from here we can check multiple options, like loadbalancer ip under services options and many more configurations
(If see error related to extension installation, close and launch vscode as

administrator)