## Not Important

Step 1:

github :

1- create repository on my local github

2- import the external repository to my local github

Using github import from website Or using command :

1- git clone --bare URL-External

2- git push --mirror URL-My local repository

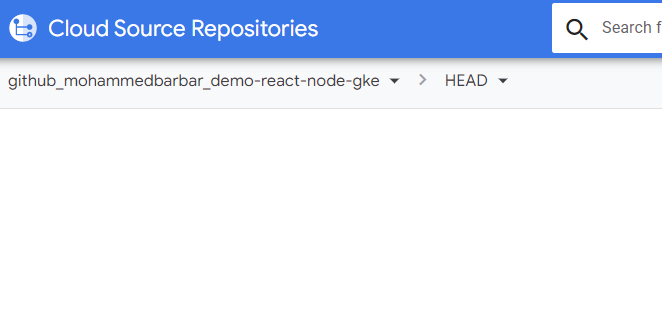
Step 2:

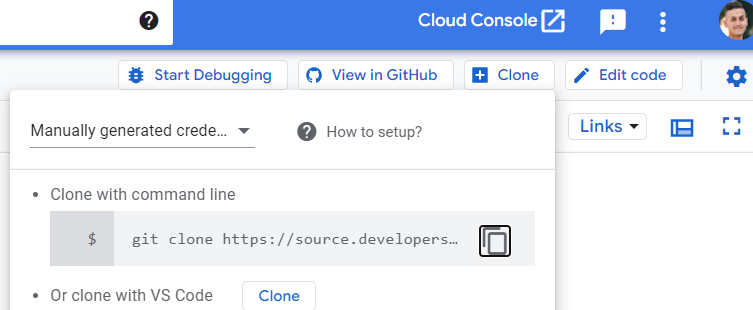
1- Create Cloud repository

2- Mirroring With my github repository

3- go to cloud shell & clone the cloud repository to my project using the command below :

git clone https://source.developers.google.com/p/devops-analiza/r/github\_mohammedbarbar\_demo-react-node-gke --branch main

(Note: I use --branch main becouse the main the repository does not shown the main branch directly but shown the head )



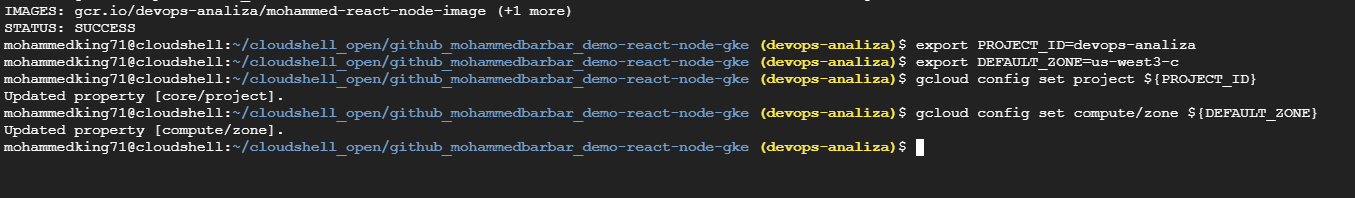
4- cd github\_mohammedbarbar\_demo-react-node-gke

5- gcloud builds submit --tag gcr.io/devops-analiza/(image-name)

Note: (image – name : Name with your choice in my example the name is mohammed-react-node-image)

Step 3 :

Build Cluster :



(Note : I worked on cloudshell\_open/my\_repo so I can make changes to the files in cloud repo)

export PROJECT\_ID=devops-analiza  
export DEFAULT\_ZONE=us-west3-c  
  
gcloud config set project ${PROJECT\_ID}  
gcloud config set compute/zone ${DEFAULT\_ZONE}



gcloud container clusters create mohammed-cluster-gke \

--project=${PROJECT\_ID} \

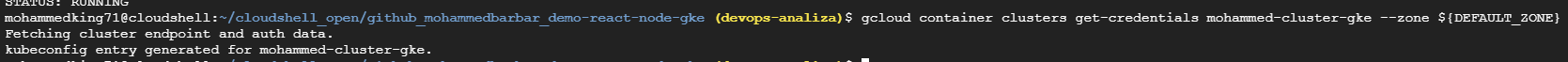
--zone=${DEFAULT\_ZONE} \

--network=default

(Note : Network here talking about VPC , I choose an existing vpc called default

3- Configure kubectl

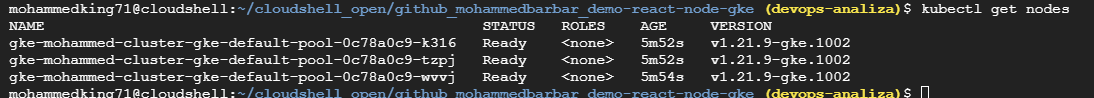
gcloud container clusters get-credentials mohammed-cluster-gke --zone ${DEFAULT\_ZONE}



4-It should now be possible to use kubectl to interact with the cluster.

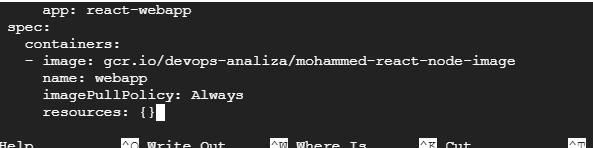
Command : kubectl get nodes

It shows your nodes in VM



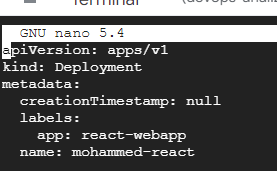
Step 4 :

1-Make changes to manifest.yml file using nano manifest.yml



We change the image to our image that we build in step 2 (5)

And also we can change the name of the service by convert the name in the yaml file

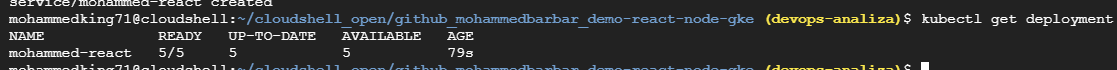


2-Now we create our service using this command

kubectl apply -f manifest.yml



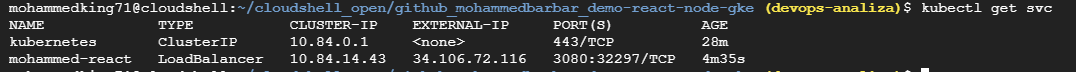
3-by using command : kubectl get deployment



We can see our service deployment with (my name service : mohammed-react)

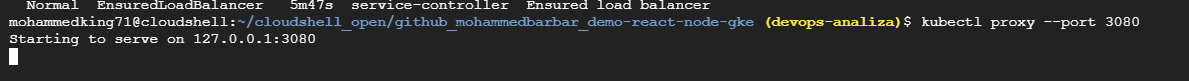
4-we get service and external ip from this comman :

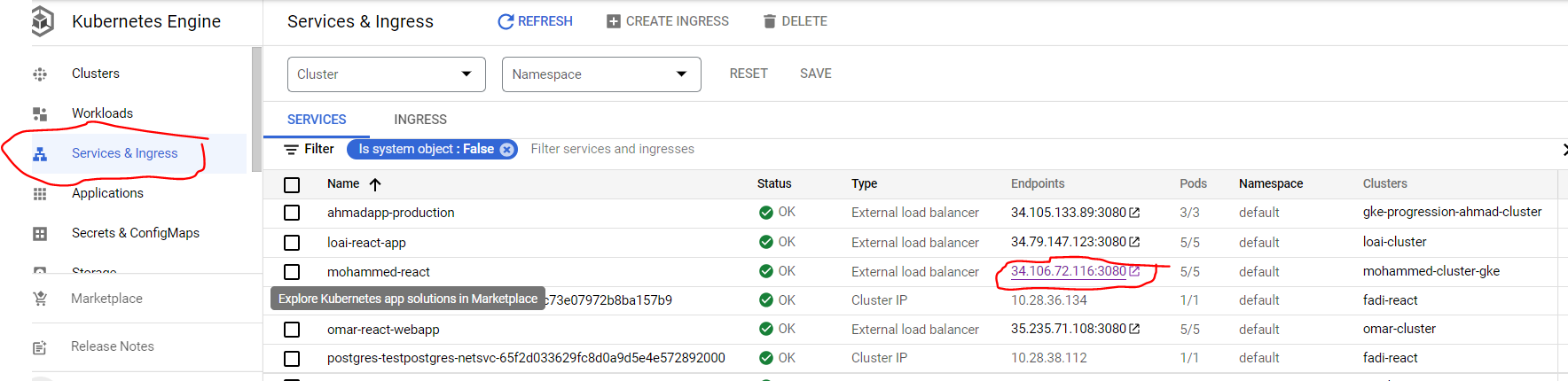
kubectl get svc



5- we run our service using this command :

kubectl proxy --port 3080



6-

Here our external Ip service running correctly .

Finally:

