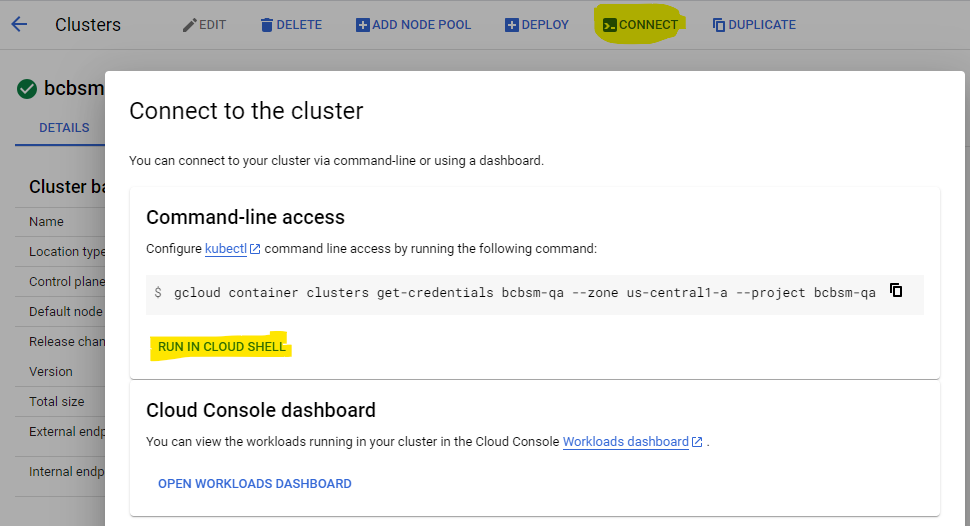
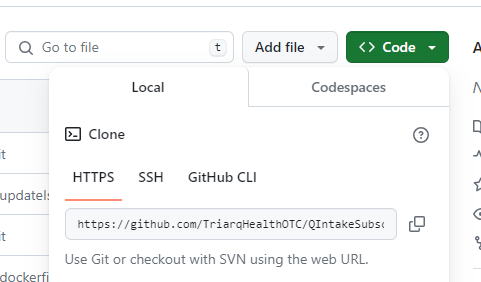
**CI/CD Deployment on Google Kubernetes Engin in GitOps style**

1. Connect the Cluster that we need to deploy the Workload



1. For the deployment we need the Docker image that is store in the GitHub Clone the Github Repository in the Cloudshell

# git clone ${repository URL}



1. Go to that directory

# Cd ${repository name}

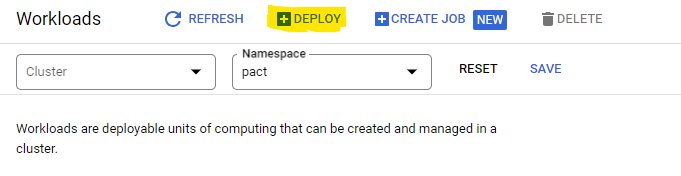
1. Check the Dockerfile is present or not. If present create the image

# docker build -t gcr.io/<project\_name>/<repo\_name>:v1 .

1. Push that image into the container registory

# docker push gcr.io/<project\_name>/<repo\_name>:v1

1. **Go to Workload and select the DEPLOY**



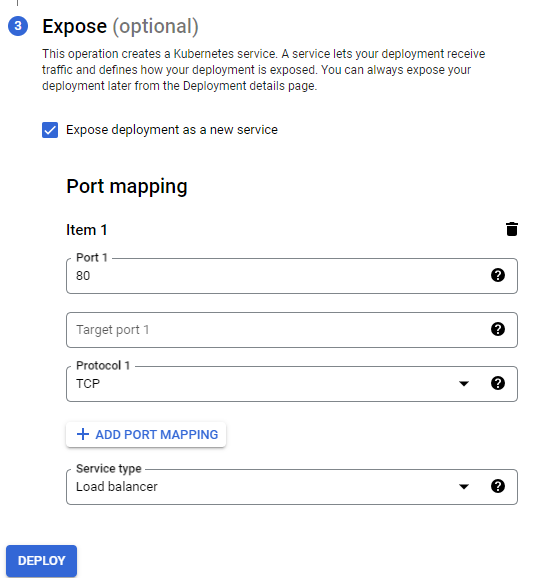
**Select the Existing container image and then select the image that we push**A screenshot of a computer

Description automatically generated

**Configuration🡪 give the name same as repo\_name to identify**

**Cluster 🡪 select the cluster that we deploy**

**Expose 🡪 Expose the Container to the NodePort**

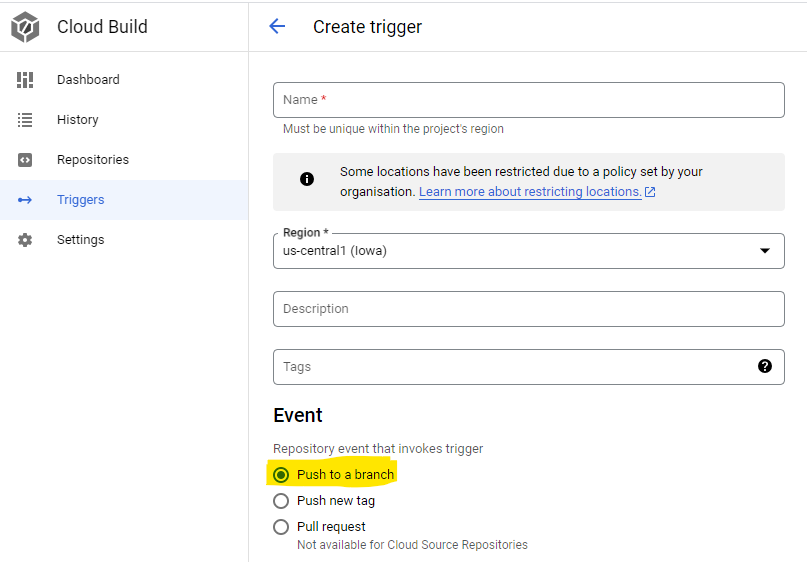
****

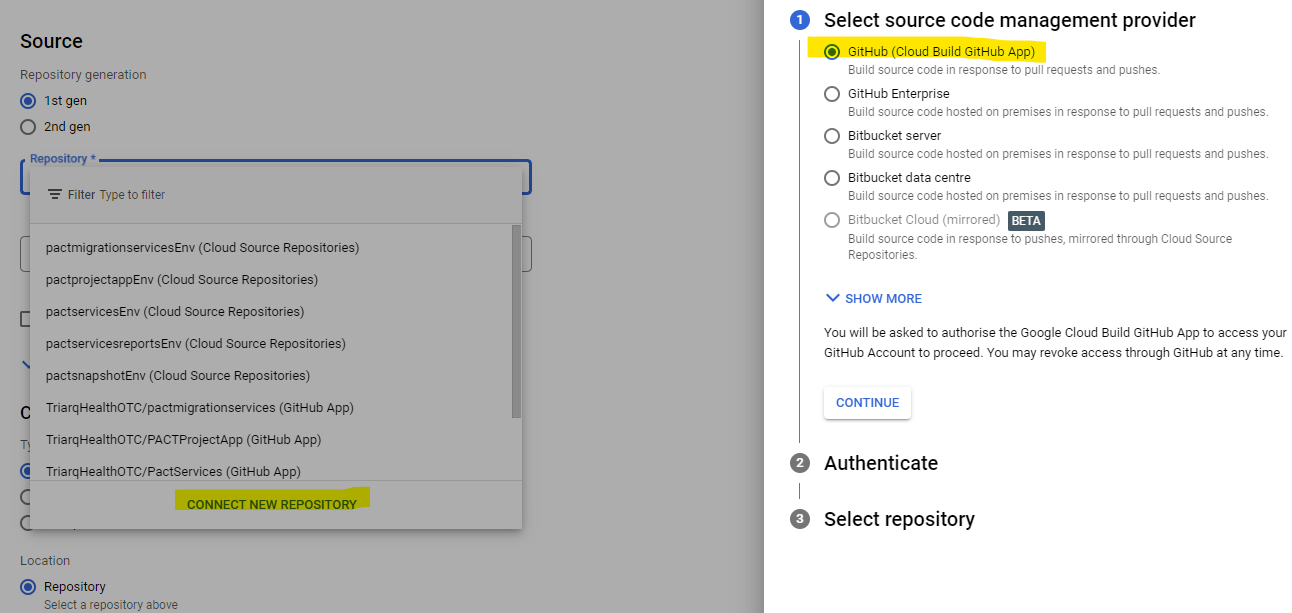
1. **In the Github repository create the three filekubernetes.yaml , kubernetes.yaml.tpl and cloudbuild.yaml**

A screenshot of a computer

Description automatically generated

1. **Go to CloudBuild to create the trigger**



**Select the repository**

Cloud Build configuration file location 🡪 cloudbuild.yaml

**Create**

1. Go to cloudshell check the **gitops.sh**

It is used for creating the **trigger** also creating the **cloud source repository**

1. change in **cloudbuild.yaml** 🡪 env (region and cluster name)
2. execute **🡪 # sh gitops.sh ${repo\_name}Env**
3. for creating the ConfigMap

**# kubectl create configmap rpapayerpathbot2-config --from-literal Port="5432" --from-literal Host="10.143.128.8"**

1. For creating the secret

**# kubectl create secret generic rpapayerpathbot2-secret --from-literal Username=postgres --from-file=Password=rpapayerpathbot2.txt**