**Documentation for deployment of code**

**Create Docker image:**

<All commands are to be executed inside the code directory>

The Cloud Shell environment variable DEVSHELL\_PROJECT\_ID automatically has your current project ID stored. The project ID is required to store images in Container Registry.

Cloud Build to build your docker image

gcloud builds submit --tag gcr.io/$DEVSHELL\_PROJECT\_ID/devops-image:v0.1 .

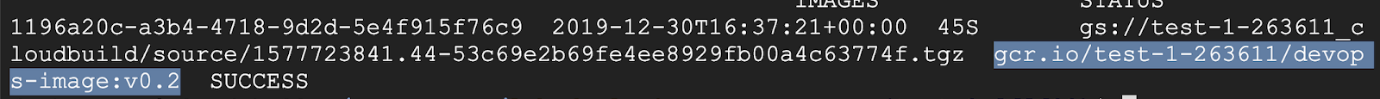
Graphical user interface, text, application, email

Description automatically generated

We can also use docker commands to create docker container

Docker commands:

docker build -t ms-image . (Docker Build)



docker run --rm -p 8080:8080 ms-image(Docker run)

docker tag ms-image gcr.io/$DEVSHELL\_PROJECT\_ID/ms-image

docker push gcr.io/$DEVSHELL\_PROJECT\_ID/ms-image

**Deploy App engine:**

In a project, an App Engine application has to be created. This is done just once using the ***gcloud app create*** command and specifying the region where you want the app to be created.

gcloud app create --region=us-central

gcloud app deploy --version=one –quiet (deploy app)

gcloud app deploy --version=two --no-promote –quiet (To deploy version two with new changes)

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Deploy Kubernetes engine:**

Graphical user interface, text, application, letter, email

Description automatically generated

You can find the Kubernetes-config.yaml file in the directory. (In file change <YOUR IMAGE PATH HERE> to created docker image path)

Text

Description automatically generated

<https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

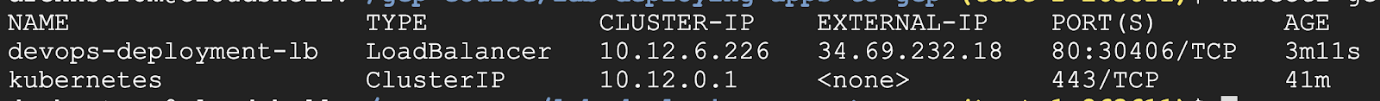
<https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/>

To deploy application:

kubectl apply -f kubernetes-config.yaml

To find the service and the external ip: (service name: ms-deployment-lb)

kubectl get services



**Using Cloud Run:**

Graphical user interface, text, application, email

Description automatically generated

**Deployment of GKE Using Terraform:**

**Link:** [**https://learn.hashicorp.com/tutorials/terraform/gke**](https://learn.hashicorp.com/tutorials/terraform/gke)

Initialize Terraform by running:

cd tfinfra

terraform init

For provisioning the GKE cluster:

terraform apply

Text

Description automatically generated

Configure kubectl:

gcloud container clusters get-credentials $(terraform output -raw kubernetes\_cluster\_name) --region $(terraform output -raw region)