**Krish Naik:** [**https://www.youtube.com/watch?v=lAAeJDcoHKY&t=150s**](https://www.youtube.com/watch?v=lAAeJDcoHKY&t=150s)

**Moez Ali:** [**https://towardsdatascience.com/deploy-machine-learning-model-on-google-kubernetes-engine-94daac85108b**](https://towardsdatascience.com/deploy-machine-learning-model-on-google-kubernetes-engine-94daac85108b)

**On GCP:**

1. **Clone REPO**

git clone <https://github.com/pycaret/pycaret-deployment-google.git>

1. **CD into the directory**

cd pycaret-deployment-google

1. **Set Project ID Environment Variable**

export PROJECT\_ID=insurance-bill-prediction

1. **Building the docker image**

docker build -t gcr.io/${PROJECT\_ID}/insurance-app:v1 .

1. **Check docker images (command gives list of local images on our machine)**

docker images

**[Uploading the container image to Google Container Registry]**

[**https://www.youtube.com/watch?v=9CDb9ZSsfV4**](https://www.youtube.com/watch?v=9CDb9ZSsfV4)

1. **Go to Container Registry and turn it on (probably first time use only):** <https://console.cloud.google.com/gcr/images/insurance-bill-prediction?project=insurance-bill-prediction&folder&organizationId>
2. **Authenticate**

gcloud auth configure-docker gcr.io

1. **Tag the image (**0ef78bb6f47c is image ID, can be checked using ‘docker images’ command**)**

docker tag 0ef78bb6f47c gcr.io/insurance-bill-prediction/insurance-app:v1

1. **Push the image**

gcloud docker –- push gcr.io/insurance-bill-prediction/insurance-app:v1

***[see DOCKER DOCUMENTATION:*** [***https://cloud.google.com/sdk/gcloud/reference/docker***](https://cloud.google.com/sdk/gcloud/reference/docker)***]***

‘’’

**EXAMPLES**

To pull the image 'gcr.io/google-containers/pause:1.0' from the docker registry, run:

gcloud docker -- pull gcr.io/google-containers/pause:1.0

Push the image 'gcr.io/example-org/example-image:latest' to our private docker registry.

gcloud docker -- push gcr.io/example-org/example-image:latest

Configure authentication, then simply use docker:

gcloud docker --authorize-only  
docker push gcr.io/example-org/example-image:latest

‘’’

1. Go to Container Registry and click REFRESH to see the deployed app under the hostname GCR.IO

**[Create cluster]**

Now that the container has been uploaded, we would need a cluster to run the container. A cluster consists of a pool of Compute Engine VM instances, running Kubernetes.

1. Set the Compute Engine zone options for the gcloud tool:

gcloud config set compute/zone us-central1

1. Creating the cluster:

gcloud container clusters create insurance-cluster --num-nodes=1

In case of error, enable API service - container.googleapis.com using below code, before creating the cluster:

gcloud services enable container.googleapis.com

[REF: <https://stackoverflow.com/questions/64537546/error-gcloud-container-clusters-create-responseerror-code-400-message-faile>]

NOTE: NUM\_NODES created are 3 (default value) instead of 1, not sure why

Maybe due to syntax change -> now NUM\_NODES (I gave num-nodes)

* Tried second time with NUM\_NODES, it showed syntax error, so not sure really.

‘’’

**NUM NODES:** <https://cloud.google.com/sdk/gcloud/reference/container/clusters/create>

Note that NUM\_NODES nodes will be created in each zone, such that if you specify --num-nodes=4 and choose one additional zone, 8 nodes will be created.

Multiple locations can be specified, separated by commas. For example:

gcloud container clusters create example-cluster  --zone us-central1-a  --additional-zones us-central1-b,us-central1-c

This flag is deprecated. Use --node-locations=PRIMARY\_ZONE,[ZONE,…] instead.

‘’’

1. Go to Menu>Kubernetes Engine >Clusters and see the created cluster

**[DEPLOY APP]**

kubectl create deployment insurance-app --image=gcr.io/${PROJECT\_ID}/insurance-app:v1

1. Exposing app to the internet:

kubectl expose deployment insurance-app --type=LoadBalancer --port 80 --target-port 8080

1. Checking the service:

kubectl get service

This may take some time. The EXTERNAL-IP shown after executing above command may show PENDING for sometime. Repeat the command after some time to see if its now up and running.