Software and Data Engineering
CSL 7090

Assignment 3: Deploying Application to Server
Instructor: Dr. Sumit Kalra

Submitted by

Ajit Kumar
M21CS017

Introduction

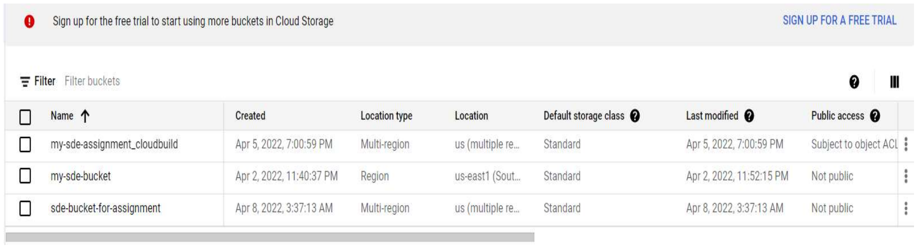
I have written a program in python for face detection using GOOGLE CLOUD VISION API. It takes the image from the current directory and google cloud storage too and detect the face inside the image and draw line around the faces.

To deploy the service , I have used kubernates engine. In kubernates engine deployment, I have make **dockerimage** . Docker image is stored into the Artifact registry. Finally I deploy this service through kubenetes to the google cloud.

For load balancing,

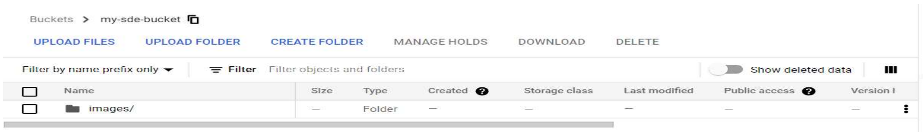
Procedure

- First created a project as **my-sde-assignment.**
- Then I have created a bucket to upload all the image that is used in our program into cloud storage bucket name as **my-sde-bucket**.



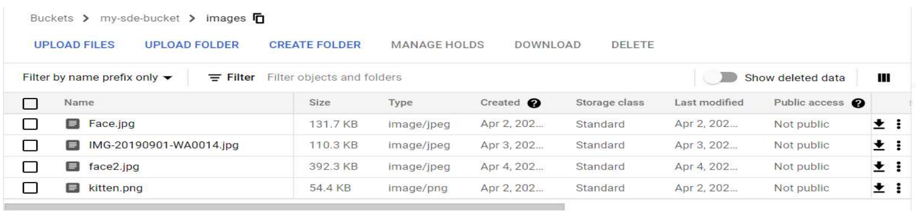
Sign up for the free trial to start using more buckets in Cloud Storage

Name	Created	Location type	Location	Default storage class	Last modified	Public access
my-sde-assignment_cloudbuild	Apr 5, 2022, 7:00:59 PM	Multi-region	us (multiple re...	Standard	Apr 5, 2022, 7:00:59 PM	Subject to object ACL
my-sde-bucket	Apr 2, 2022, 11:40:37 PM	Region	us-east1 (Sout...	Standard	Apr 2, 2022, 11:52:15 PM	Not public
sde-bucket-for-assignment	Apr 8, 2022, 3:37:13 AM	Multi-region	us (multiple re...	Standard	Apr 8, 2022, 3:37:13 AM	Not public



Buckets > my-sde-bucket

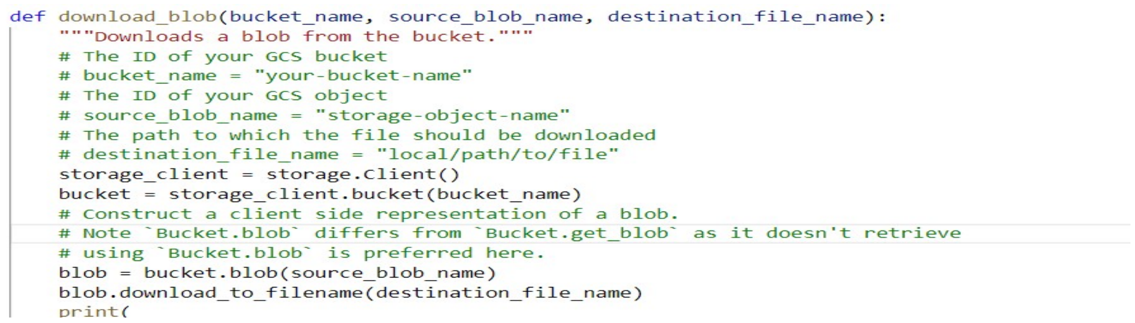
Name	Size	Type	Created	Storage class	Last modified	Public access	Version I
images/	—	Folder	—	—	—	—	—



Buckets > my-sde-bucket > images

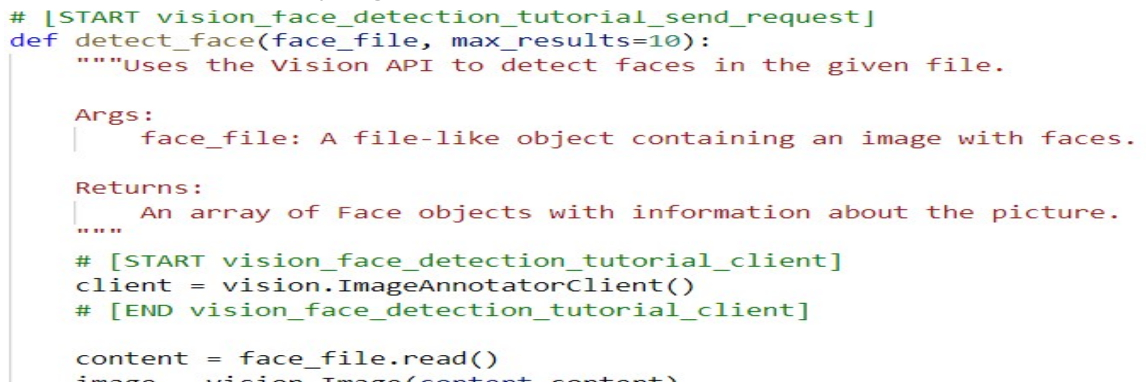
Name	Size	Type	Created	Storage class	Last modified	Public access
Face.jpg	131.7 KB	image/jpeg	Apr 2, 202...	Standard	Apr 2, 202...	Not public
IMG-20190901-WA0014.jpg	110.3 KB	image/jpeg	Apr 3, 202...	Standard	Apr 3, 202...	Not public
face2.jpg	392.3 KB	image/jpeg	Apr 4, 202...	Standard	Apr 4, 202...	Not public
kitten.png	54.4 KB	image/png	Apr 2, 202...	Standard	Apr 2, 202...	Not public

- Then I have written program for my requirement.
- Below screen shot is program for downloading the image from cloud storage(bucket).



```
def download_blob(bucket_name, source_blob_name, destination_file_name):
    """Downloads a blob from the bucket."""
    # The ID of your GCS bucket
    # bucket_name = "your-bucket-name"
    # The ID of your GCS object
    # source_blob_name = "storage-object-name"
    # The path to which the file should be downloaded
    # destination_file_name = "local/path/to/file"
    storage_client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    # Construct a client side representation of a blob.
    # Note `Bucket.blob` differs from `Bucket.get_blob` as it doesn't retrieve
    # using `Bucket.blob` is preferred here.
    blob = bucket.blob(source_blob_name)
    blob.download_to_filename(destination_file_name)
    print(
```

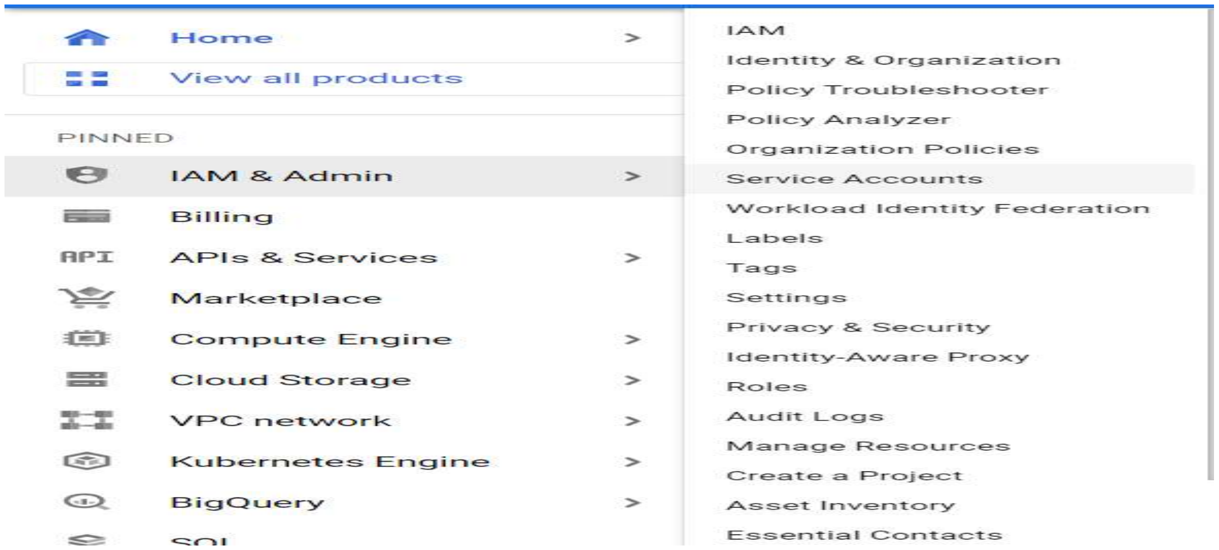
- Below screen shot is program for detect face.

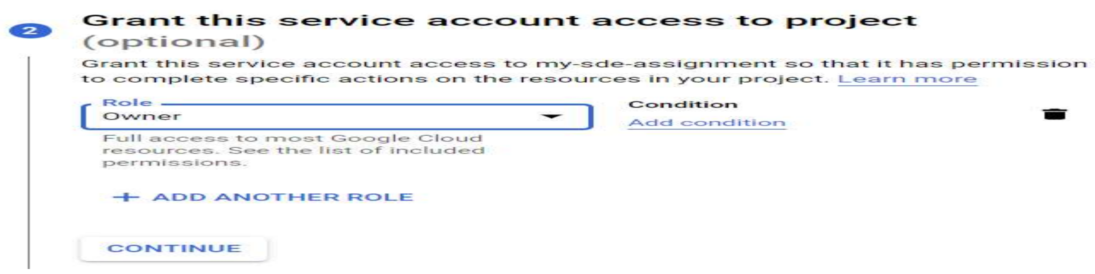


```
# [START vision_face_detection_tutorial_send_request]
def detect_face(face_file, max_results=10):
    """Uses the Vision API to detect faces in the given file.

    Args:
        face_file: A file-like object containing an image with faces.

    Returns:
        An array of Face objects with information about the picture.
    """
    # [START vision_face_detection_tutorial_client]
    client = vision.ImageAnnotatorClient()
    # [END vision_face_detection_tutorial_client]

    content = face_file.read()
    image = vision.Image(content=content)
```

- To select the vision API in google cloud:
  a. First select the my project as my-sde-assignment.
  b. Next clone the sample repository
  c. To use the vision API, our app need to authenticate its identity to the vision service.

d. Created a service account to our API request.





2 Grant this service account access to project (optional)

Grant this service account access to my-sde-assignment so that it has permission to complete specific actions on the resources in your project. Learn more

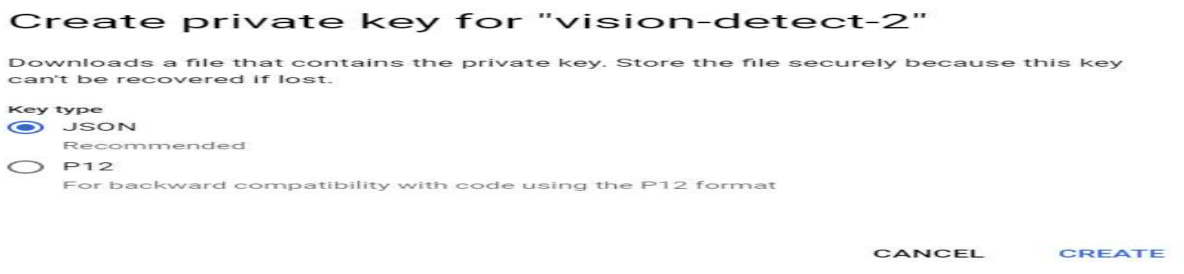
Role: Owner

Full access to most Google Cloud resources. See the list of included permissions.

Condition: Add condition

+ ADD ANOTHER ROLE

CONTINUE



Create private key for "vision-detect-2"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format

CANCEL CREATE

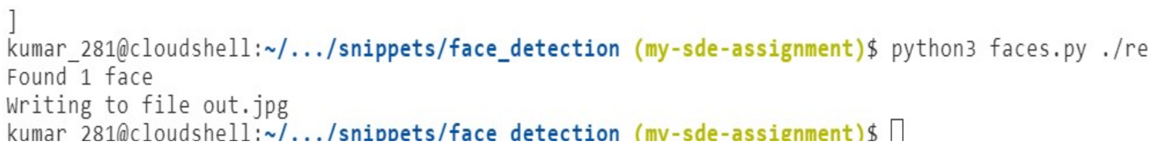
e. Then I created service account key in same directory as vision-detect-2 and set the key as default credentials. To do this I have used following command:

```
gcloud iam service-accounts keys \
create key.json --iam-account \
vision-detect-2@my-sde-assignment.iam.gserviceaccount.com

export GOOGLE_APPLICATION_CREDENTIALS=key.json
```

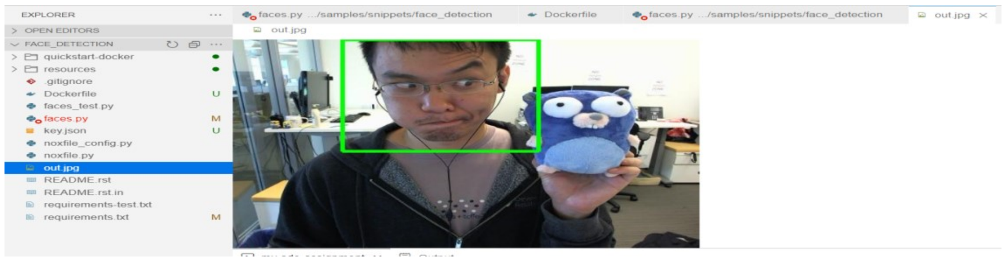
- Then to install all the requirement for the program from requirement.txt ,used the following command:

pip3 install -r requirements.txt

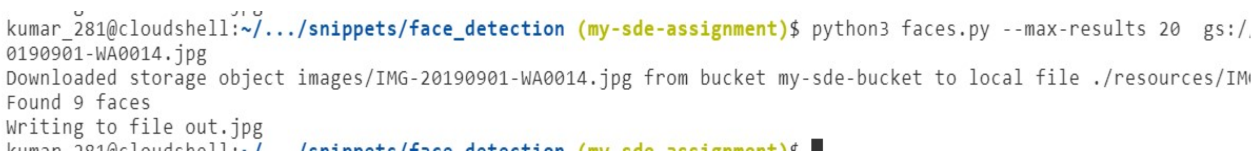
- Then finally run the program:
  a. This is taking the image from the same directory:



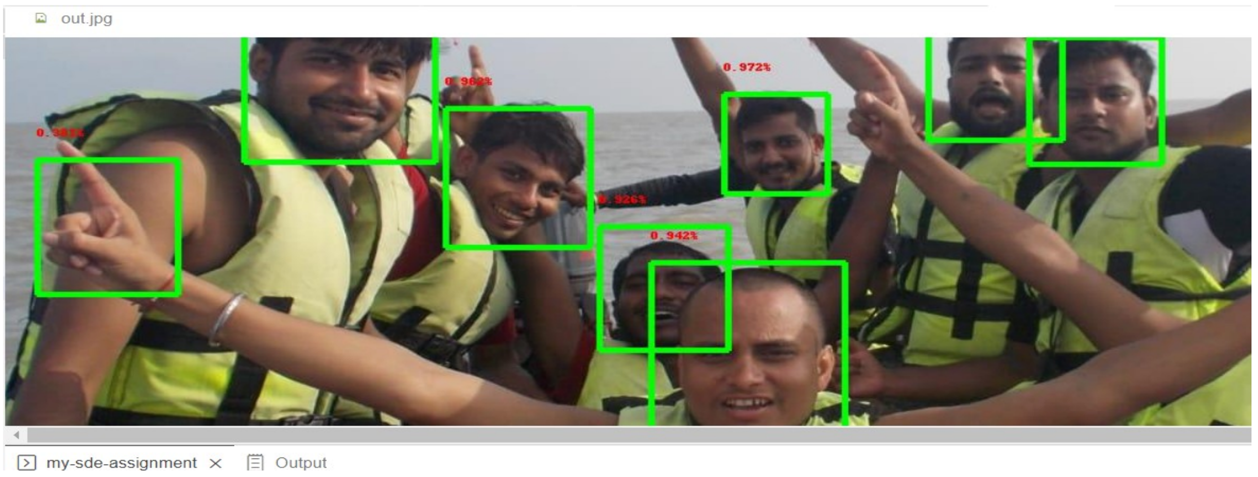
```
]
kumar_281@cloudshell:~/.../snippets/face_detection (my-sde-assignment)$ python3 faces.py ./re
Found 1 face
Writing to file out.jpg
kumar_281@cloudshell:~/.../snippets/face_detection (my-sde-assignment)$ 
```



  b. This is image taking from the google cloud storage:

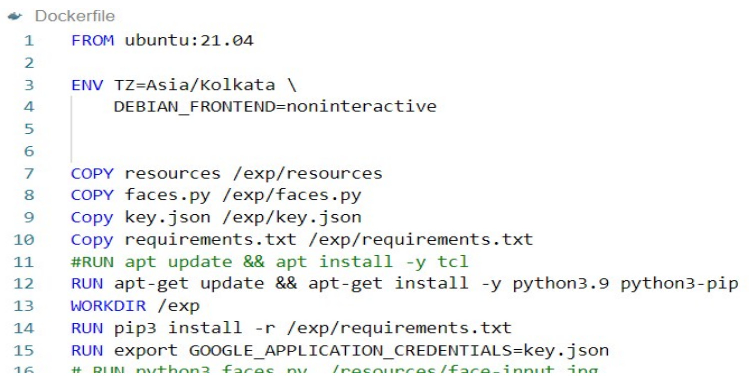


```
kumar_281@cloudshell:~/.../snippets/face_detection (my-sde-assignment)$ python3 faces.py --max-results 20  gs:/
0190901-WA0014.jpg
Downloaded storage object images/IMG-20190901-WA0014.jpg from bucket my-sde-bucket to local file ./resources/IM
Found 9 faces
Writing to file out.jpg
kumar_281@cloudshell:~/.../snippets/face_detection (my-sde-assignment)$ 
```



Deployment

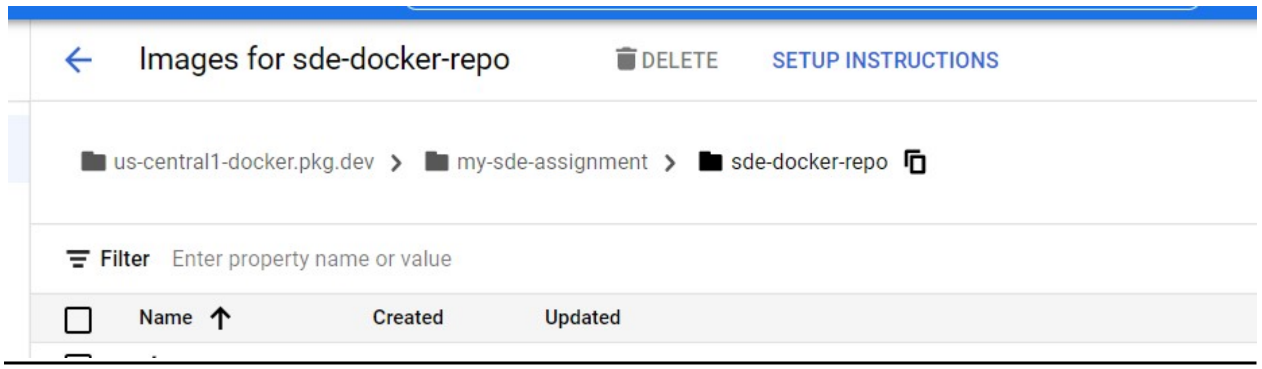
- For the deployment I have used the kubernetes , So I have created docker file.



Dockerfile

```
FROM ubuntu:21.04

ENV TZ=Asia/Kolkata \
    DEBIAN_FRONTEND=noninteractive


COPY resources /exp/resources
COPY faces.py /exp/faces.py
Copy key.json /exp/key.json
Copy requirements.txt /exp/requirements.txt
#RUN apt update && apt install -y tcl
RUN apt-get update && apt-get install -y python3.9 python3-pip
WORKDIR /exp
RUN pip3 install -r /exp/requirements.txt
RUN export GOOGLE_APPLICATION_CREDENTIALS=key.json
# RUN python3 faces.py ./resources/face-input.jpg
```

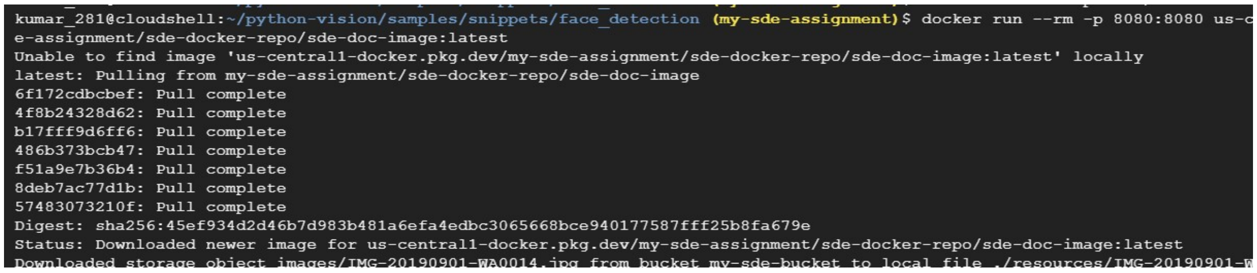
- Then I created image for above docker file:
  a. First I have created a artifact registry name as sde-docker-repo by using following command
     gcloud artifacts repositories create sde-docker-repo \
     --repository-format=docker \
     --location=us-central1 \
     --description="Docker repository"
  b. Next, I have created docker image by using following command:
     docker build -t REGION-docker.pkg.dev/${my-sde-assignment}/sde-docker-repo/sde-doc-image:latest



Images for sde-docker-repo

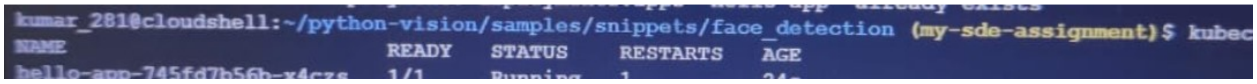
DELETE
SETUP INSTRUCTIONS
us-central1-docker.pkg.dev > my-sde-assignment > sde-docker-repo
Filter Enter property name or value
Name
Created
Updated

  c. Then I run my docker image to check my image formation occur correctly or not. I found my image is working ,For proof I have attached the following screenshot:



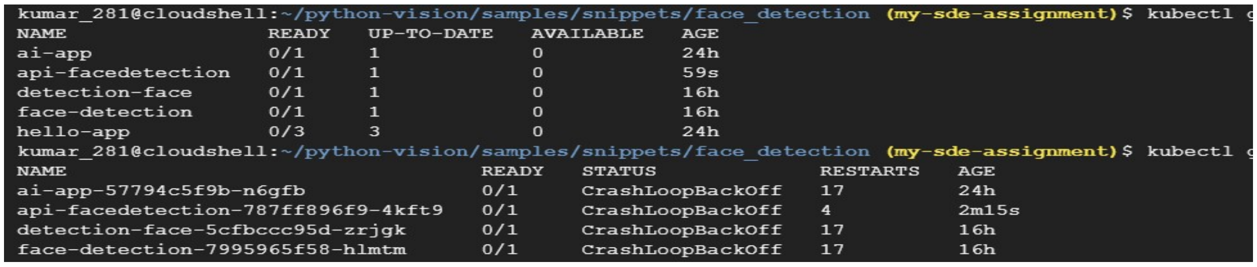
```
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ docker run --rm -p 8080:8080 us-c
e-assignment/sde-docker-repo/sde-doc-image:latest
Unable to find image 'us-central1-docker.pkg.dev/my-sde-assignment/sde-docker-repo/sde-doc-image:latest' locally
latest: Pulling from my-sde-assignment/sde-docker-repo/sde-doc-image
6f172cdbcbef: Pull complete
4f8b24328d62: Pull complete
b17fff9d6ff6: Pull complete
486b373bcb47: Pull complete
f51a9e7b36b4: Pull complete
8deb7ac77d1b: Pull complete
57483073210f: Pull complete
Digest: sha256:45ef934d2d46b7d983b481a6efa4edbc3065668bce940177587fff25b8fa679e
Status: Downloaded newer image for us-central1-docker.pkg.dev/my-sde-assignment/sde-docker-repo/sde-doc-image:latest
Downloaded storage object images/IMG-20190901-WA0014.jpg from bucket my-sde-bucket to local file ./resources/IMG-20190901-W
```

- Next I have created Google Kubernate Engine name as **sde-assignment-cluster-1** and then connect to to GKE cluster.
- To create deployment, I have used the following command: **kubectl create deployment hello-app --image=us-central1-docker.pkg.dev/my-sde-assignment/sde-docker-repo/sde-doc-image:latest**
- Next I checked , my deployment is ready or not.



```
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubec
NAME                          READY   STATUS    RESTARTS   AGE
hello-app-745fd7b56b-x4czs    1/1     Running   1          24
```

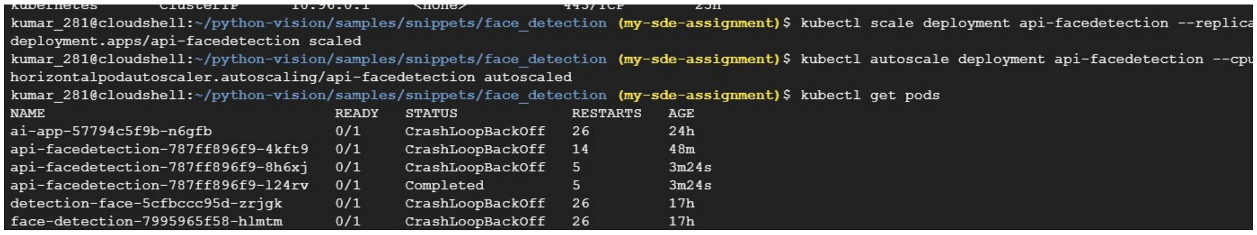
- On the time of deployment, It was in ready state but after some time it was not ready so I tried to do more deployment but it happens same with every deployment and it's reason **CrashLoopBackOff** as you can see in following screenshot :



```
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubectl g
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
ai-app               0/1     1            0           24h
api-facedetection    0/1     1            0           59s
detection-face       0/1     1            0           16h
face-detection       0/1     1            0           16h
hello-app            0/3     3            0           24h
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubectl g
NAME                                 READY   STATUS             RESTARTS   AGE
ai-app-57794c5f9b-n6gfb              0/1     CrashLoopBackOff   17         24h
api-facedetection-787ff896f9-4kft9   0/1     CrashLoopBackOff   4          2m15s
detection-face-5cfbccc95d-zrjgk      0/1     CrashLoopBackOff   17         16h
face-detection-7995965f58-hlmtm      0/1     CrashLoopBackOff   17         16h
```

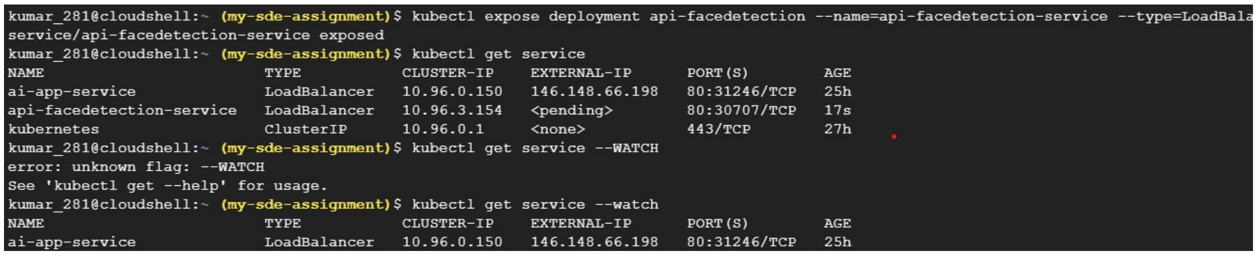
- Recently I have done one deployment name as api-face-detection and it also gone to unready state.

- To set the baseline number of deployment replica to 3 and to create HorizonatalPodScaler resources , I have used the following command:



```
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubectl scale deployment api-facedetection --replica
deployment.apps/api-facedetection scaled
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubectl autoscale deployment api-facedetection --cpu
horizontalpodautoscaler.autoscaling/api-facedetection autoscaled
kumar_281@cloudshell:~/python-vision/samples/snippets/face_detection (my-sde-assignment)$ kubectl get pods
NAME                                 READY   STATUS             RESTARTS   AGE
ai-app-57794c5f9b-n6gfb              0/1     CrashLoopBackOff   26         24h
api-facedetection-787ff896f9-4kft9   0/1     CrashLoopBackOff   14         48m
api-facedetection-787ff896f9-8h6xj   0/1     CrashLoopBackOff   5          3m24s
api-facedetection-787ff896f9-l24rv   0/1     Completed          5          3m24s
detection-face-5cfbccc95d-zrjgk      0/1     CrashLoopBackOff   26         17h
face-detection-7995965f58-hlmtm      0/1     CrashLoopBackOff   26         17h
```

- To generate kubernetes service for the api-face-detection used the following command:



```
kumar_281@cloudshell:~ (my-sde-assignment)$ kubectl expose deployment api-facedetection --name=api-facedetection-service --type=LoadBala
service/api-facedetection-service exposed
kumar_281@cloudshell:~ (my-sde-assignment)$ kubectl get service
NAME                        TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)        AGE
ai-app-service              LoadBalancer   10.96.0.150   146.148.66.198   80:31246/TCP   25h
api-facedetection-service   LoadBalancer   10.96.3.154   <pending>        80:30707/TCP   17s
kubernetes                  ClusterIP      10.96.0.1     <none>           443/TCP        27h
kumar_281@cloudshell:~ (my-sde-assignment)$ kubectl get service --WATCH
error: unknown flag: --WATCH
See 'kubectl get --help' for usage.
kumar_281@cloudshell:~ (my-sde-assignment)$ kubectl get service --watch
NAME                        TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)        AGE
ai-app-service              LoadBalancer   10.96.0.150   146.148.66.198   80:31246/TCP   25h
```

- In the above screenshot , we got the external ip-address as **34.135.216.101** and we can see that its type is load balancer.

REFERENCES

https://cloud.google.com/vision/docs/face-tutorial?hl=en_US

YouTube Link

https://www.youtube.com/watch?v=lRTU0GlO_T8