Build A Usecase Smart Security For Homes Using IBM Cloud

Rishabh Maheshwari
19BCY10145

Build a use-case Smart Security for Homes using IBM cloud.

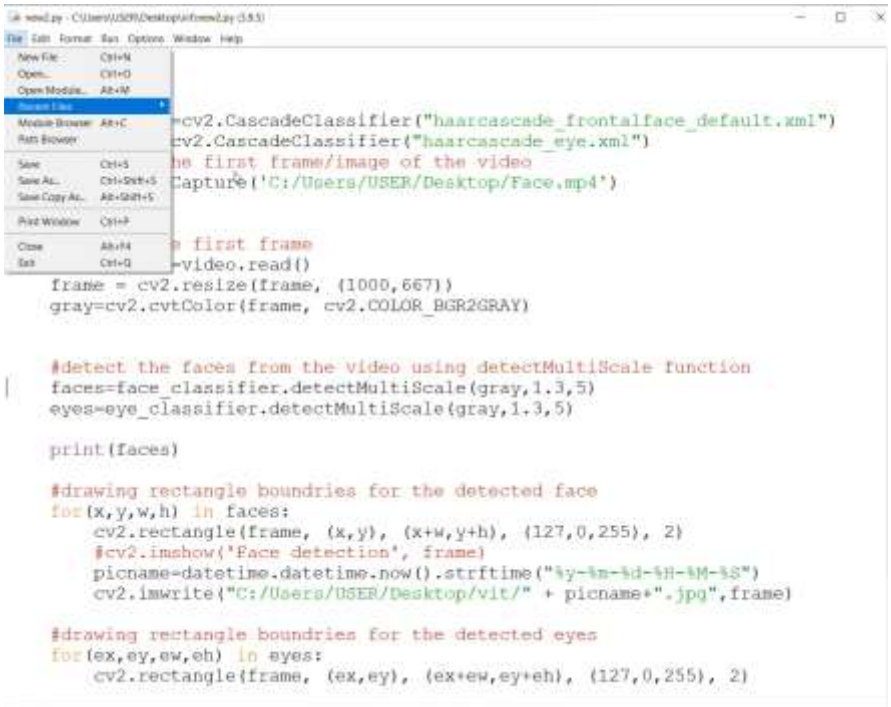
The features of the project are:

- capture the image if any person is detected in video streaming using python code
- send the image to IBM cloud object storage
- send the image URL to cloudant DB
- develop a mobile app to display the image and control the doors
- write a python code to receive the commands and control the doors (servo rotation)

1. The python program for Detecting a face, Capturing the Images and Storing them locally on the PC.

```
import cv2

face_classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier = cv2.CascadeClassifier("haarcascade_eye.xml")
#It will read the first frame/image of the video
video=cv2.VideoCapture('pexels-cottonbro-8090198')
```

```
while True:
    #capture the first frame
    check,frame=video.read()
    frame = cv2.resize(frame, (1920, 1080))
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    #detect the faces from the video using detectMultiScale function
    faces = face_classifier.detectMultiScale(gray,1.3,5)
    eyes = eye_classifier.detectMultiScale(gray,1.3,5)

    print(faces)

    #drawing rectangle boundries for the detected face
    for(x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)

    #drawing rectangle boundries for the detected eyes
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(frame, (ex,ey), (ex+ew,ey+eh), (127,0,255), 2)
        cv2.imshow('Face detection', frame)

    #waitKey(1)- for every 1 millisecond new frame will be captured
    Key=cv2.waitKey(25)
    if Key==ord('q'):
        #release the camera
        video.release()
        #destroy all windows
        cv2.destroyAllWindows()
        break
```

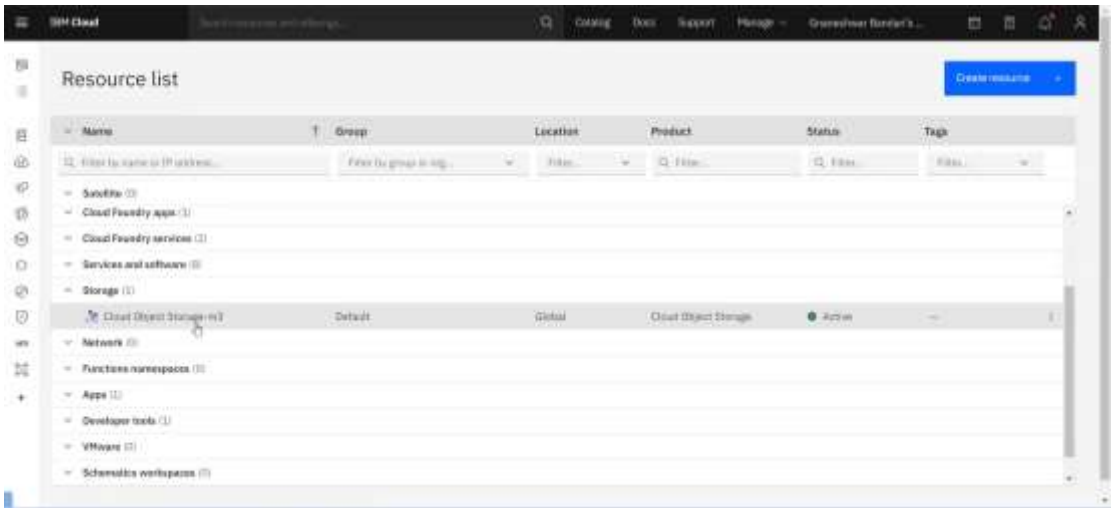


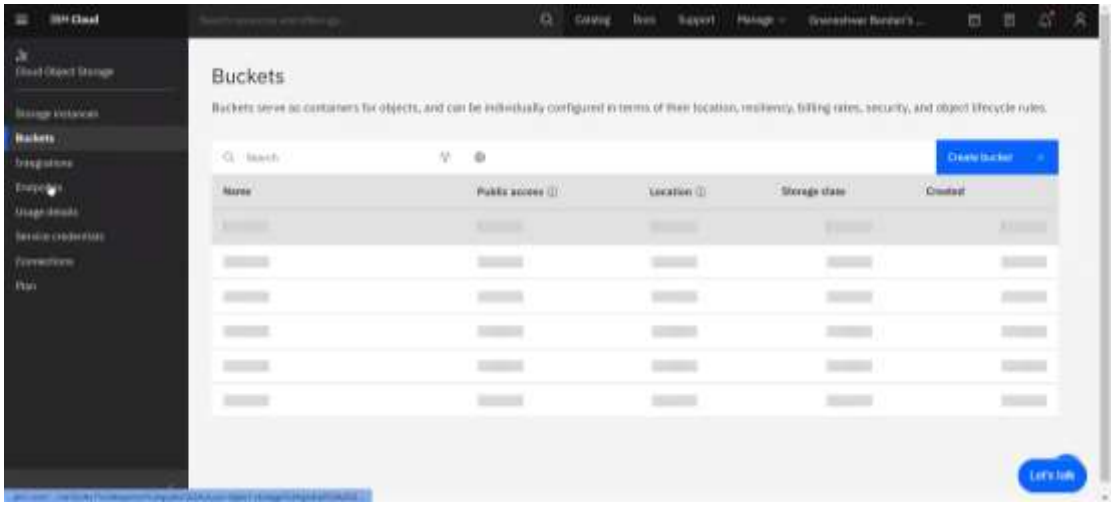
2. Python program for sending data and files to IBM Cloudant Database.

```
from ibmcloudant.cloudant_v1 import CloudantV1, Document
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
import random

authenticator = BasicAuthenticator('apikey-v2-16u3crmdpkghhxefdikvpuxoh5fwexrmuu
                                   'b0ab119f45dbe6255eabb978e7e2f0e1')

service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-16u3crmdpkghhxefdikvpuxoh5fwexrmuup5f
'''
response = service.put_database(db='products1', partitioned=True).get_result()
print(response)'''
'''r = random.randint(0,1000)
products_doc = Document(
  id="small-appliances:1000042"+str(r),
  type="product",
  productid="1000042",
  random = r,
  brand="Salter",
  name="Digital Kitchen Scales",
  description="Slim Colourful Design Electronic Cooking Appliance for Home / Kit
  price=14.99,
  image="assets/img/Opmanghhew.jpg")

response = service.post_document(db='products1', document=products_doc).get_resu

print(response)'''

response = service.get_document(
  db='products1',
  doc_id='small-appliances:1000042'
```

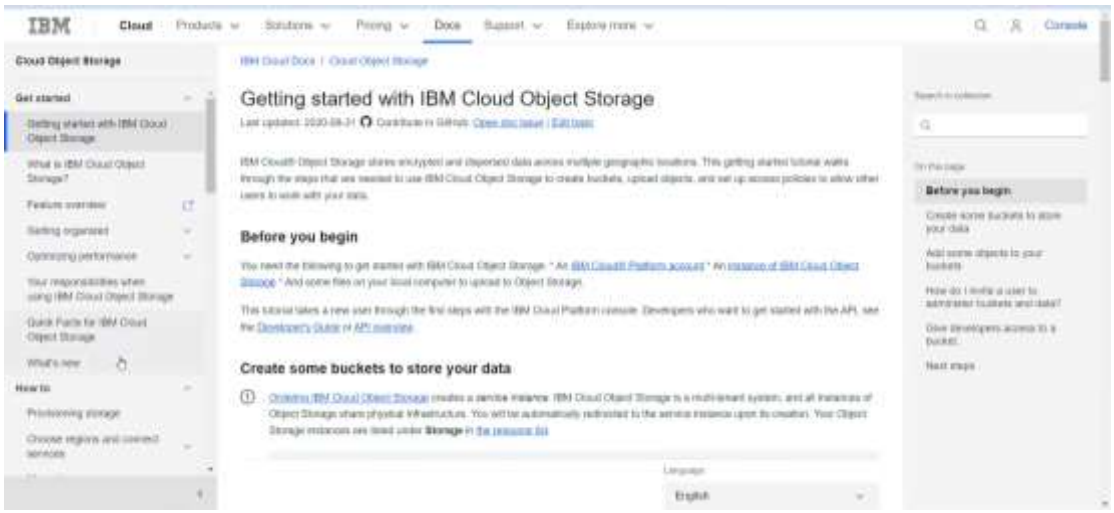
```
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-16u3crmdpkghhxefdikvpuxoh5fwexrmuup5f
'''
response = service.put_database(db='products1', partitioned=True).get_result()
print(response)'''
'''r = random.randint(0,1000)
products_doc = Document(
  id="small-appliances:1000042"+str(r),
  type="product",
  productid="1000042",
  random = r,
  brand="Salter",
  name="Digital Kitchen Scales",
  description="Slim Colourful Design Electronic Cooking Appliance for Home / Kit
  price=14.99,
  image="assets/img/Opmanghhew.jpg")

response = service.post_document(db='products1', document=products_doc).get_resu

print(response)'''

response = service.get_document(
  db='products1',
  doc_id='small-appliances:1000042'
).get_result()

print(response)
```

3. Open Cloudant Object Storage Service and go to Buckets.

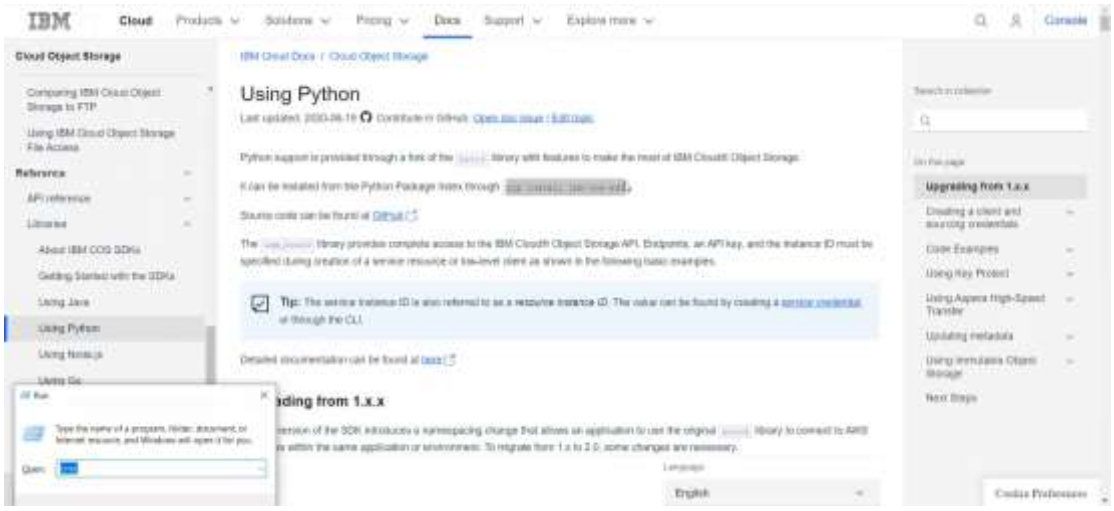




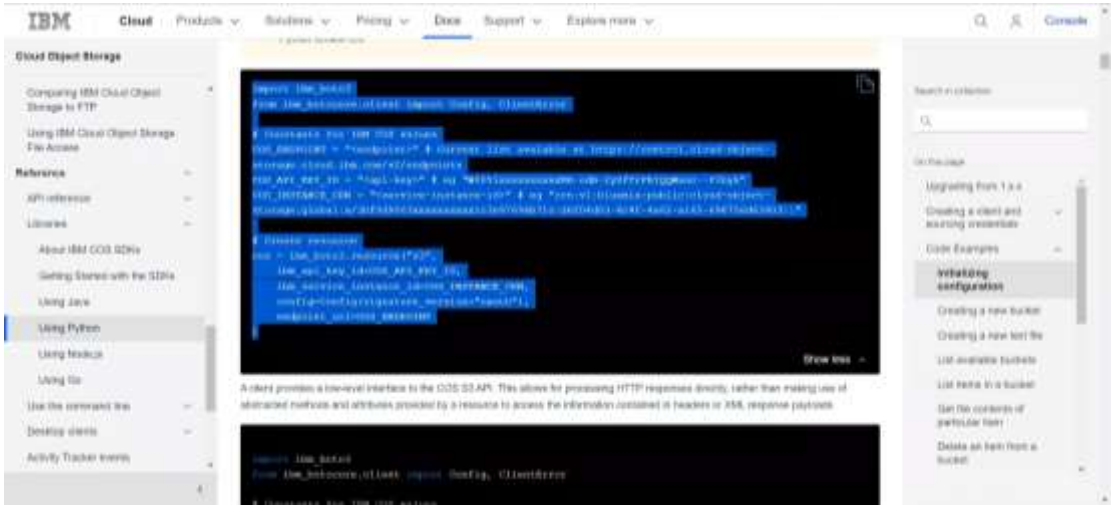
4. Open IBM Cloud Documents. Go to Libraries and open Using Python Docs.



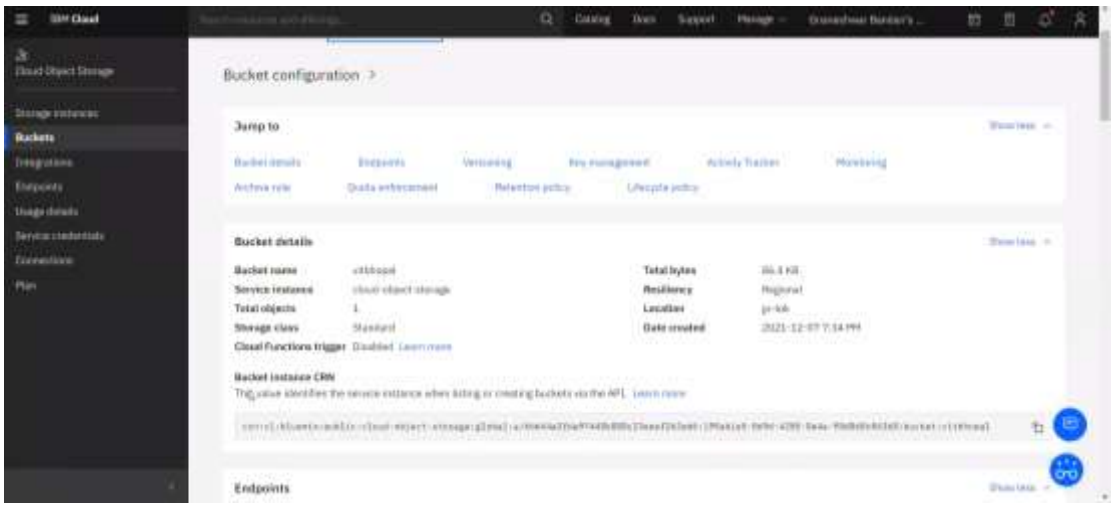
5. Open Command Prompt and install ibm-cos-sdk using pip.



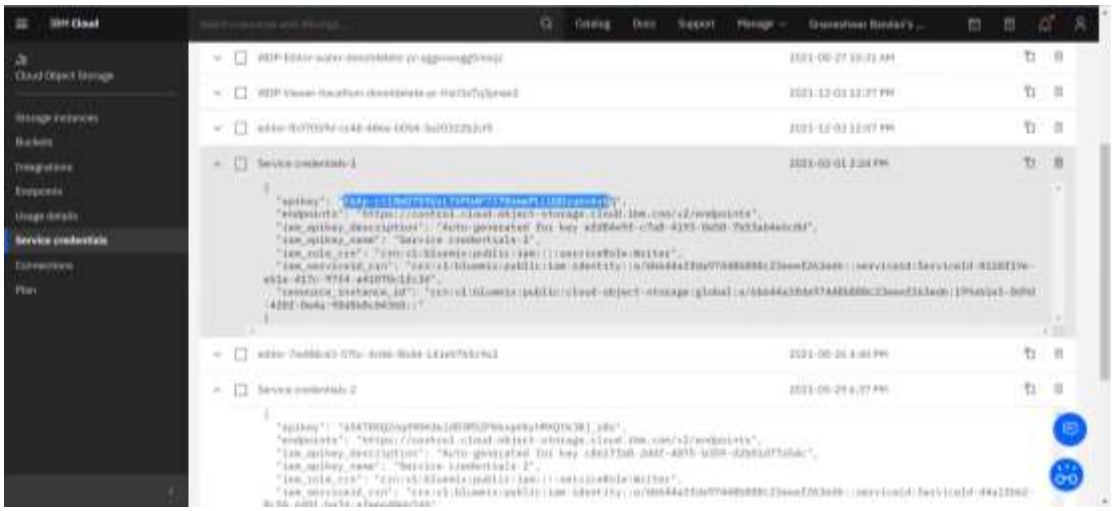
6. Copy the code for Installing Configurations and paste them in a new python file.



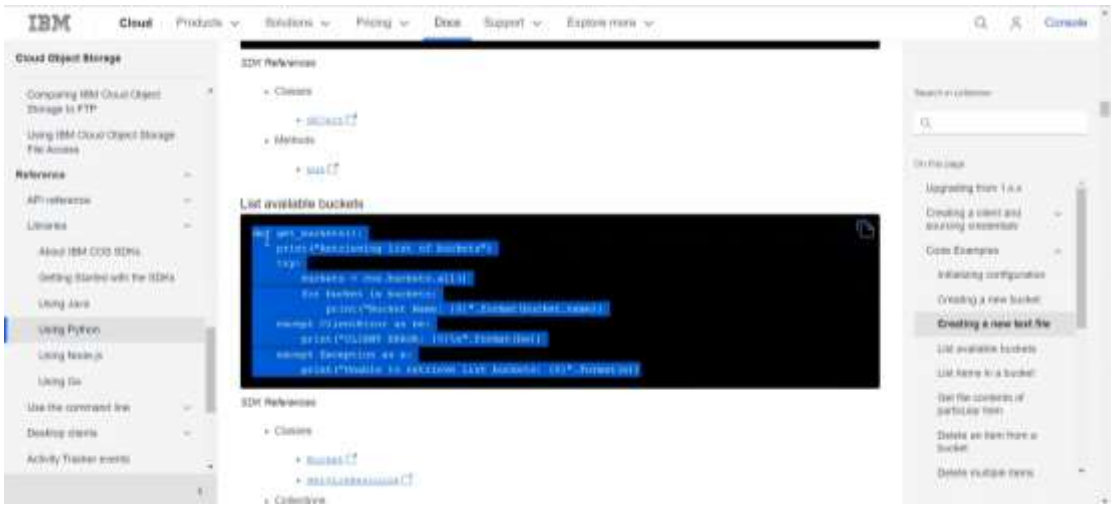
7. Go to Buckets and get the bucket Instance CRN (creating buckets using API) and paste them inside the python code.



8. Get the API key from service Credentials created before.

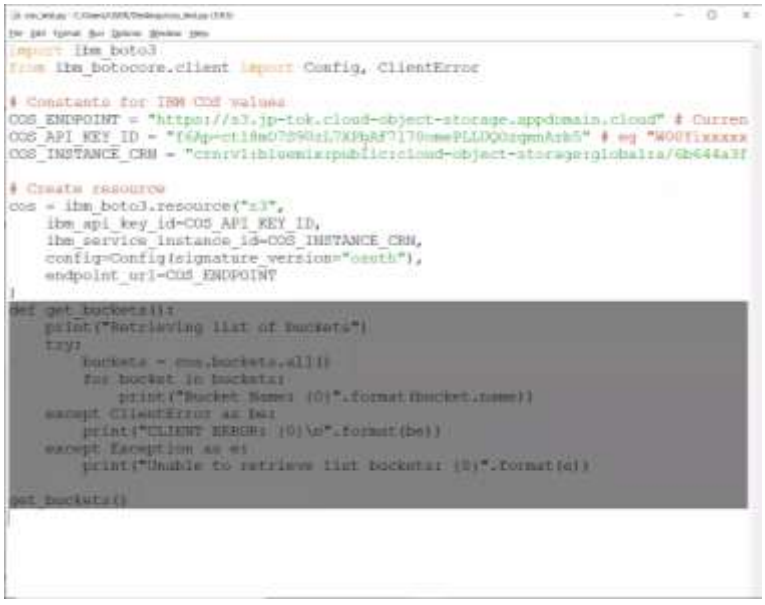


9. For completing the code we just need to create new text file whose python code is available in same documents section from before.



10. After the code has been completed run the code and the output would show all Databases stored in the Cloudant Object Database.

11. To run a multi-part upload copy the code and add it to the Python code.



```
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part uplo
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))
```

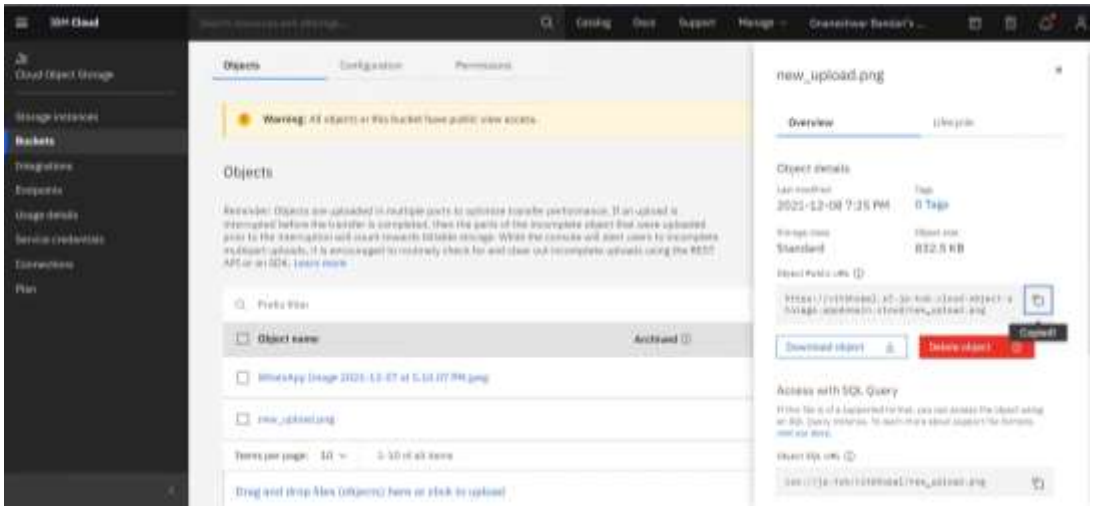
12. The Output will upload a new file to the Database.

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/USER/Desktop/cos_test.py ==========
Retrieving list of buckets
Bucket Name: 4529c9a7-5ac1-4e75-a40e-16210fb0012e
Bucket Name: esternshiplot
Bucket Name: f8965964-ce75-4eab-be2c-c5bfeeeee635
Bucket Name: gnaneshwar
Bucket Name: gnitocrop-donotdelete-pr-qbb4ebkuvlv5yu
Bucket Name: hacathon-donotdelete-pr-frsn5a7q3pnan2
Bucket Name: maskdetect
Bucket Name: motormaintenance-donotdelete-pr-qkjxkrprrurn4r
Bucket Name: notebookdeployment-donotdelete-pr-eglzg9owkcosve
Bucket Name: vitbhopal
Bucket Name: water-donotdelete-pr-agpavvxqgqsmoqz
>>>
========== RESTART: C:/Users/USER/Desktop/cos_test.py ==========
Starting file transfer for new_upload.png to bucket: vitbhopal

Transfer for [redacted]pload.png Complete!

>>>
```

13. The multi_part_upload must be specified with the database followed by the File name followed by the complete file location on the pc.

```
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part uplo
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))

multi_part_upload("vitbhopal", "new_upload.png", "C:/Users/USER/Desktop/new.png"
```

14. The properties of the file in the Database will contain a public accessible URL. Open the URL to access the Image Captured.



15. The control a Servo motor to open a gate on sensing a face you have to make a flow as given below. The output will be as shown.