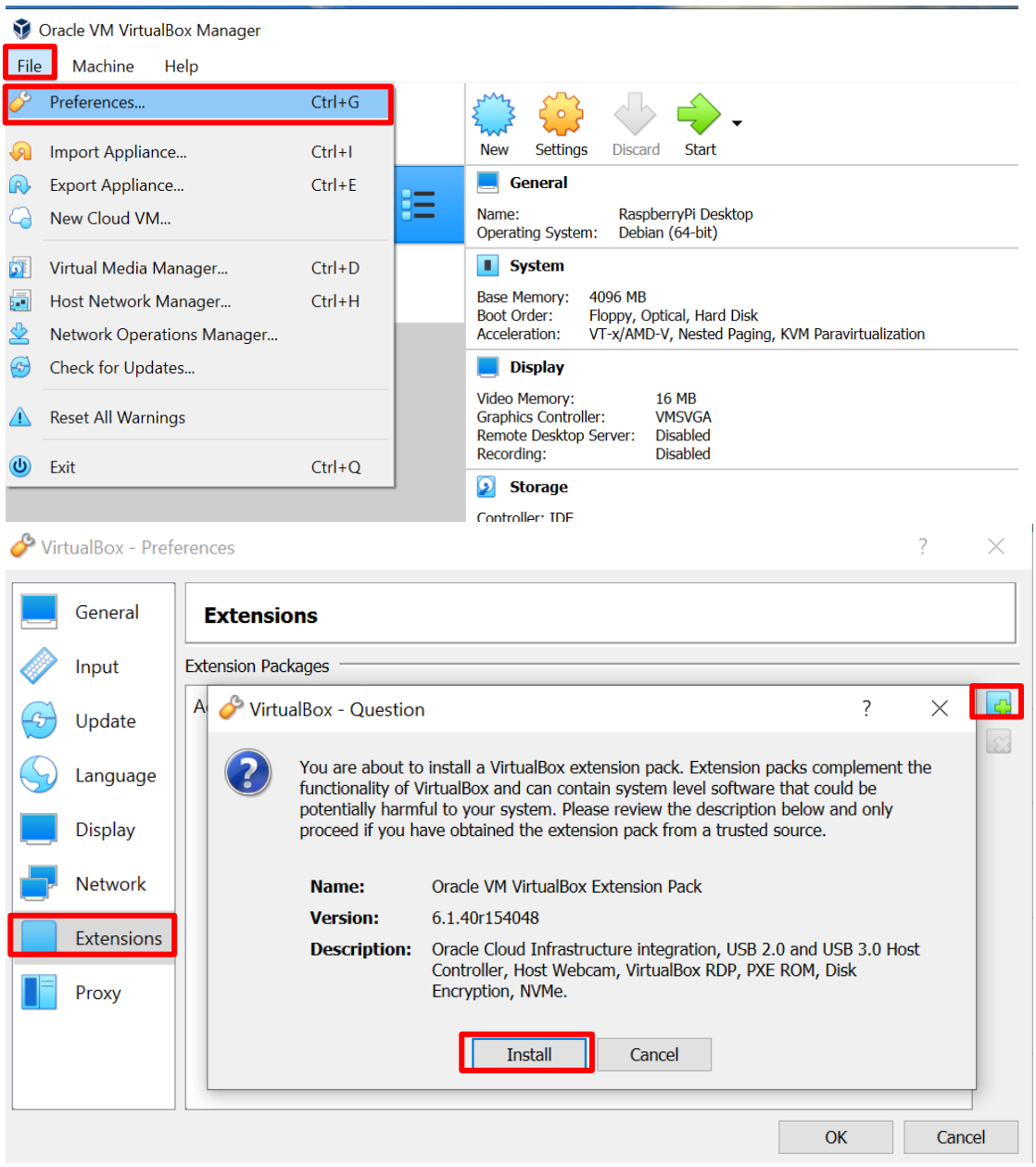
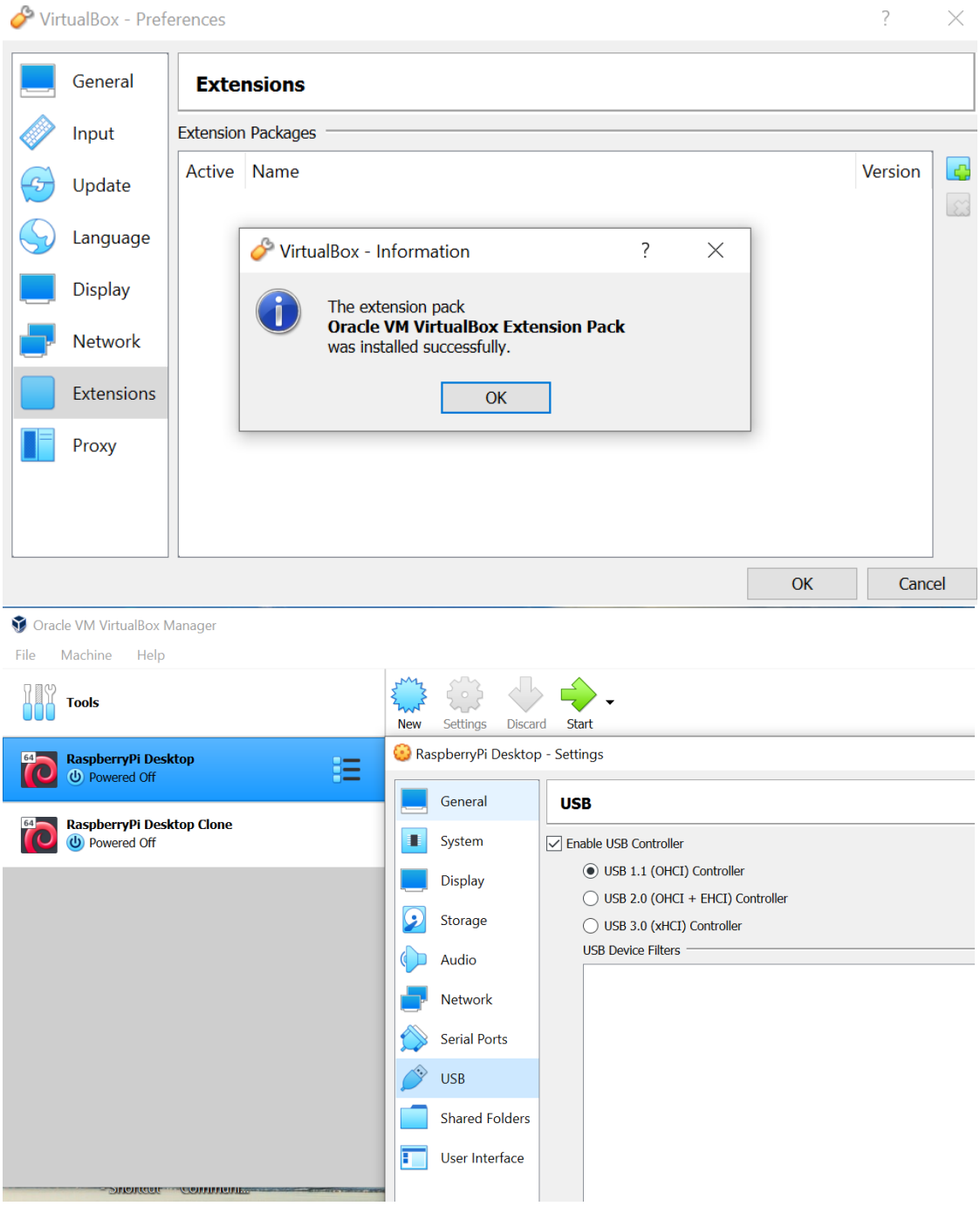
Week12: Homework1: Facial Recognition on Raspberry Pi with AWS Rekognition

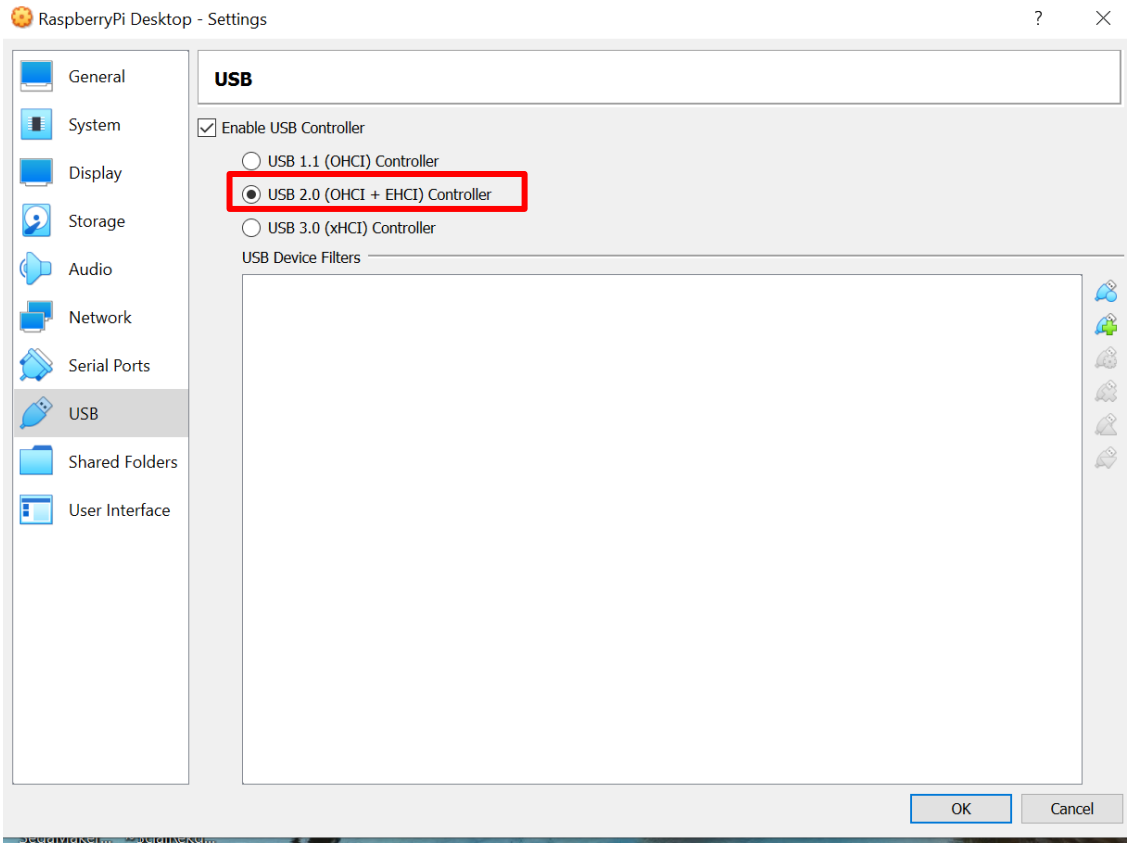
1. Install extension pack for VirtualBox

   a. Download extension pack for VirtualBox from this website [Downloads – Oracle VM VirtualBox]
   b. Open VirtualBox, follow the following instructions to install extension pack as below:





Note: The virtual machine I am using is "RaspberryPi Desktop". It is a Linux Debian(64bit) VM, it works fine.



Reference link: [How to Install VirtualBox Extension Pack (lifewire.com)]

2. Attach webcam to your virtual machine

   In this project, we will use the integrated camera on my laptop.

   a. Start virtual machine "RaspberryPi Desktop"

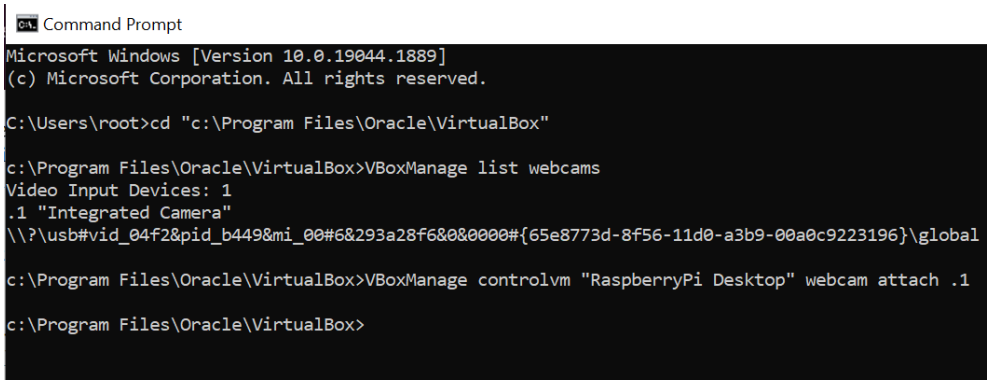
      Keep virtual machine "RaspberryPi Desktop" running

   b. Launch "Command Prompt" on your guest OS (means my laptop, windows 10)

      C:\Users\root>**cd "c:\Program Files\Oracle\VirtualBox"**

      c:\Program Files\Oracle\VirtualBox>**VBoxManage list webcams**

      c:\Program Files\Oracle\VirtualBox>**VBoxManage controlvm "RaspberryPi Desktop" webcam attach .1**

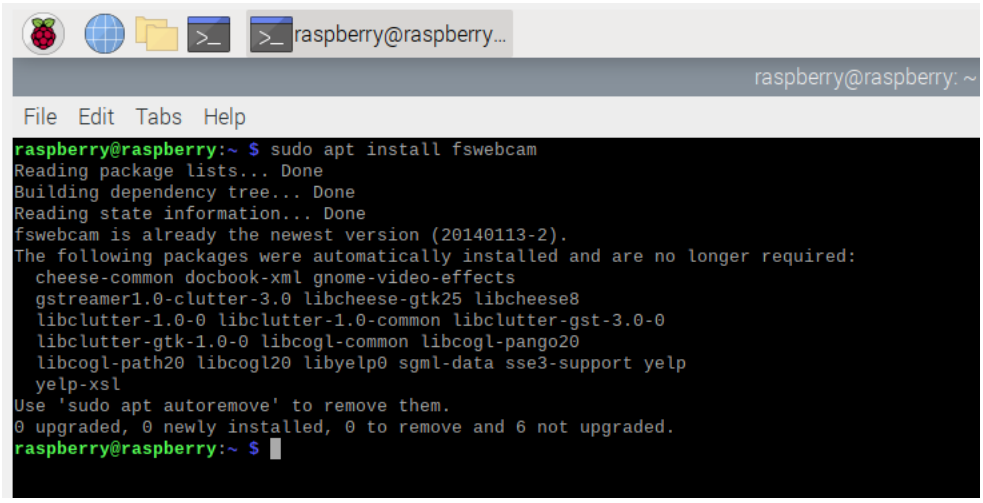


* Replace "RaspberryPi DeskTop" with your virtual machine name.

c. Back to the virtual machine “RaspberryPi Desktop” terminal, verify whether webcam works

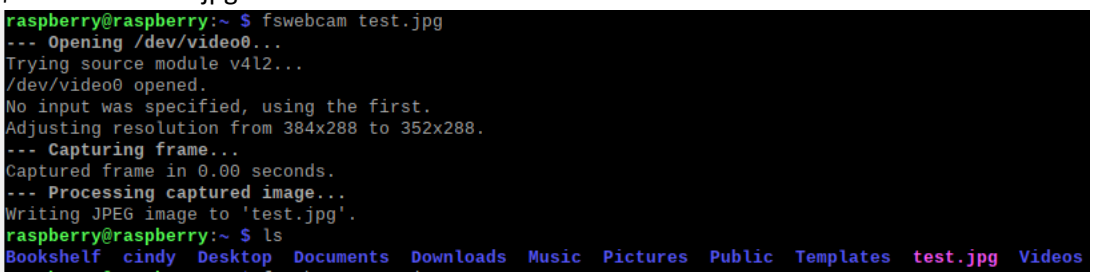
Install fswebcam

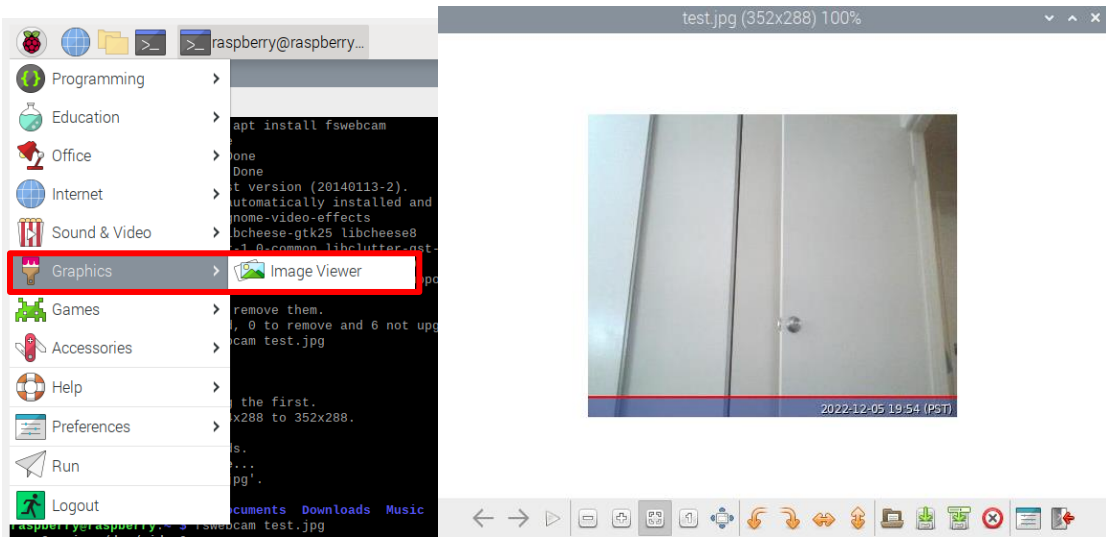
$ sudo apt install fswebcam



Test fswebcam

$ fswebcam test.jpg





3. Implementation

a. Install Boto3 and OpenCV

$ pip install boto3

```
raspberry@raspberry: ~
File  Edit  Tabs  Help
raspberry@raspberry:~ $ pip install boto3
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting boto3
  WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ProtocolError('Connection a
borted.', RemoteDisconnected('Remote end closed connection without response'))': /simple/boto3/boto3-1.26.24-py3-none-any.whl
  Downloading https://www.piwheels.org/simple/boto3/boto3-1.26.24-py3-none-any.whl (129 kB)
     |████████████████████████████████| 129 kB 168 kB/s
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading https://www.piwheels.org/simple/s3transfer/s3transfer-0.6.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 312 kB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading https://www.piwheels.org/simple/jmespath/jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting botocore<1.30.0,>=1.29.24
  Downloading https://www.piwheels.org/simple/botocore/botocore-1.29.24-py3-none-any.whl (10.2 MB)
     |████████████████████████████████| 10.2 MB 673 kB/s
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3/dist-packages (from botocore<1.30.0,>=1.29.24->boto3) (1.26.5)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3/dist-packages (from botocore<1.30.0,>=1.29.24->boto3) (2.8.1)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.26.24 botocore-1.29.24 jmespath-1.0.1 s3transfer-0.6.0
raspberry@raspberry:~ $
```

$ pip install opencv-python

```
raspberry@raspberry:~ $ pip install opencv-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-python
  Downloading opencv-python-4.6.0.66.tar.gz (90.3 MB)
     |████████████████████████████████| 90.3 MB 304 bytes/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: numpy>=1.17.3 in /usr/lib/python3/dist-packages (from opencv-python) (1.19.5)
Building wheels for collected packages: opencv-python
  Building wheel for opencv-python (PEP 517) ... done
  Created wheel for opencv-python: filename=opencv_python-4.6.0.66-cp39-cp39-linux_i686.whl size=22028738 sha256=fa6c18e910b130c0174641b16acb9a57
06199292db9545d29f80a12fa27a77bf
  Stored in directory: /home/raspberry/.cache/pip/wheels/6c/3a/b0/162197b99d01e5d1a44096c7392a6bf8ae182a4ee9a85ef9af
Successfully built opencv-python
Installing collected packages: opencv-python
Successfully installed opencv-python-4.6.0.66
raspberry@raspberry:~ $
```

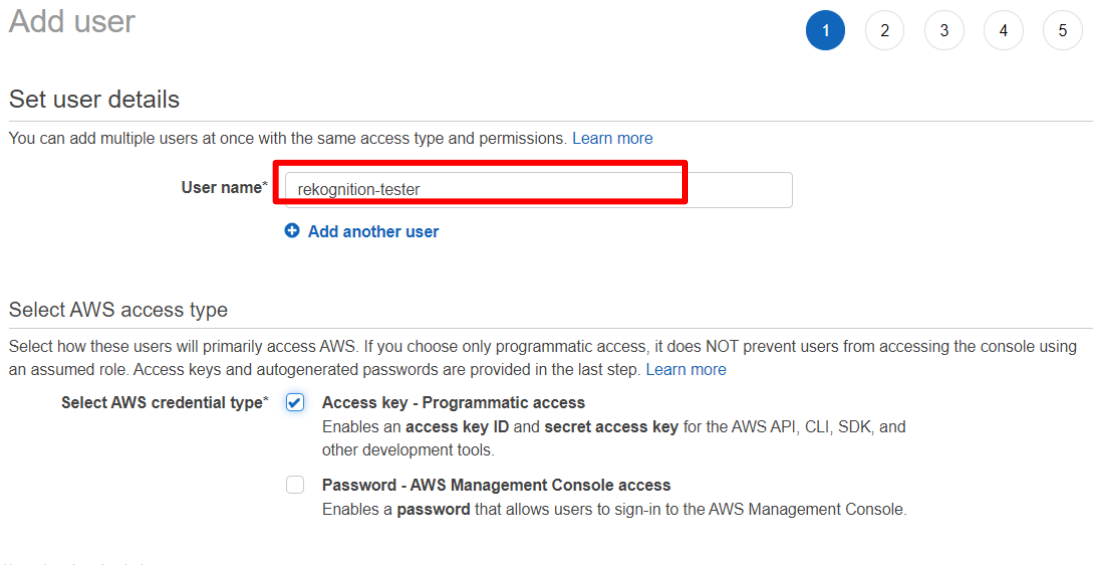
Verify that opencv-python is installed successfully

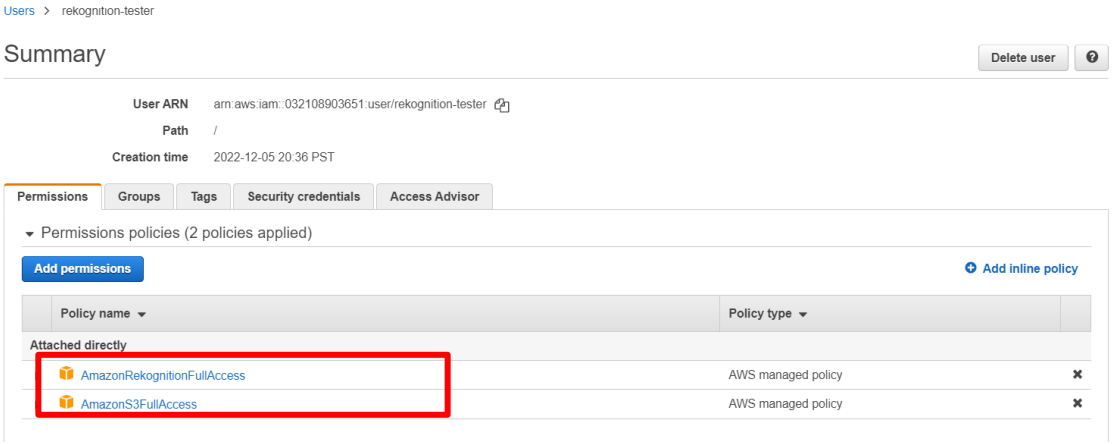
$ python3

>>> import cv2

```
raspberry@raspberry:~ $ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> quit
Use quit() or Ctrl-D (i.e. EOF) to exit
>>> quit()
raspberry@raspberry:~ $
```

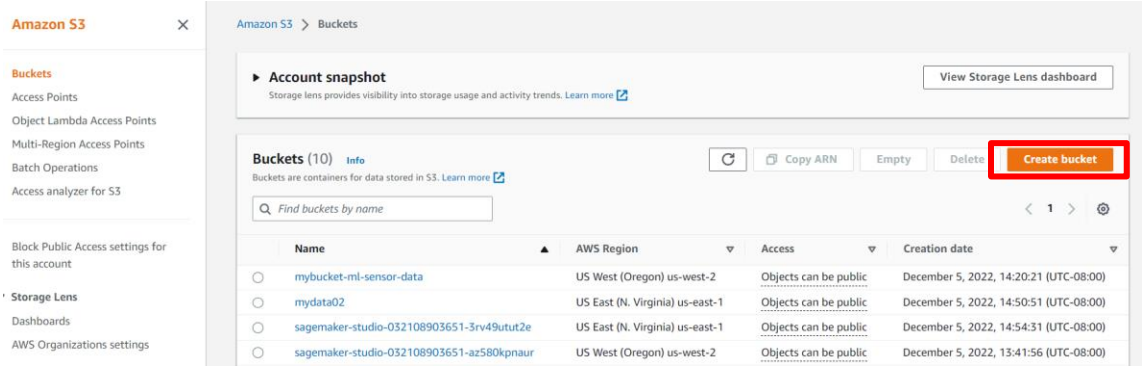
b. Create an AWS IAM user with all RekognitionFullAccess perimission

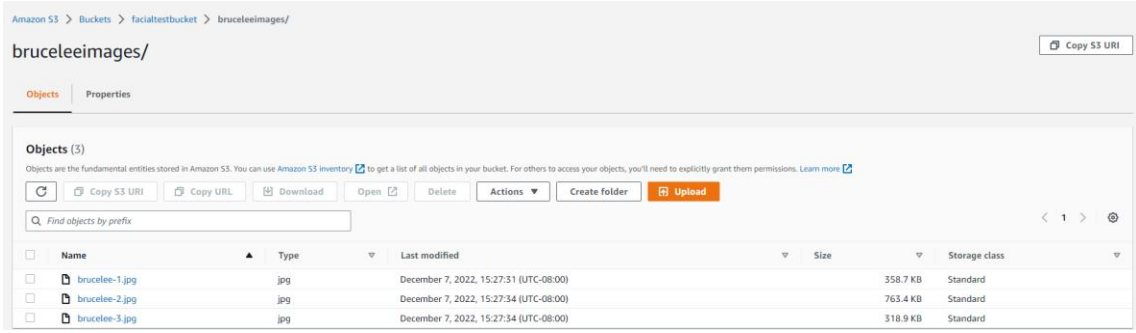


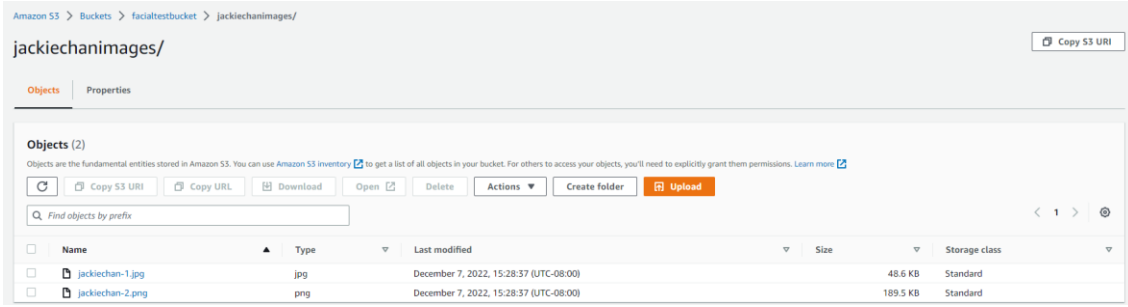


Remember that download .csv file which contains access Key ID and secret access key. We need to use this info in our program.

c. Create an AWS S3 bucket by clicking this link [S3 Management Console (amazon.com)]
I uploaded 2 folders of 2 people's images. These images are used as reference images.







d. Implement Codes

For Indexing, create facial_indexing.py:

```
import boto3

s3_client = boto3.client(
    's3',
    aws_access_key_id='AKIAQO6OW7TR74PGM6OU',# add the aws access key
    aws_secret_access_key='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'# add the aws secret access key
)

collectionId='facialindex' #collection name
rek_client=boto3.client('rekognition',
    aws_access_key_id='AKIAQO6OW7TR74PGM6OU',# add the aws access key
    aws_secret_access_key='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',# add the aws secret access key
    region_name='us-west-1',)# add the region here

bucket = 'facialtestbucket' #S3 bucket name
all_objects = s3_client.list_objects_v2(Bucket =bucket )

#delete existing collection if it exists
list_response=rek_client.list_collections(MaxResults=2)
if collectionId in list_response['CollectionIds']:
    rek_client.delete_collection(CollectionId=collectionId)

#create a new collection
rek_client.create_collection(CollectionId=collectionId)

#add all images in current bucket to the collections #use folder names as the labels
for content in all_objects['Contents']:
    collection_name,collection_image =content['Key'].split('/')
    if collection_image:
        label = collection_name
        print('indexing: ',label)
        image = content['Key']
        index_response=rek_client.index_faces(CollectionId=collectionId,
                Image={'S3Object':{'Bucket':bucket,'Name':image}},
                ExternalImageId=label,
                MaxFaces=1,
                QualityFilter="AUTO",
                DetectionAttributes=['ALL'])
        print('FaceId: ',index_response['FaceRecords'][0]['Face']['FaceId'])
~
"facial_indexing.py" [dos] 39L, 1668B
```

For Matching, create a facial_matching.py:

```
import time
import boto3
import cv2

# open camera
cap = cv2.VideoCapture(0)

collectionId='facialindex' #collection name
rek_client=boto3.client('rekognition',
aws_access_key_id='AKIAQO6OW7TR74PGM6OU',# add the aws access key
aws_secret_access_key='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',# add the aws secret access key
region_name='us-west-1',)

n=1
while True:
    time.sleep(2)
    #milli = int(round(time.time() * 1000))
    # set dimensions
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 2560)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 1440)
    image = 'image'+str(n)+'.jpg'
    n=n+1
    # take frame
    ret, frame = cap.read()
    # write frame to file
    cv2.imwrite(image, frame)

    print('captured '+image)
    with open(image, 'rb') as image:
        try: #match the captured images against the indexed faces
            match_response =rek_client.search_faces_by_image(CollectionId=collectionId, Image={'Bytes':image.read()}, MaxFaces=1, FaceMatchThreshold=85)
            if match_response['FaceMatches']:
                print('Hello, ',match_response['FaceMatches'][0]['Face']['ExternalImageId'])
                print('Similarity: ',match_response['FaceMatches'][0]['Similarity'])
                print('Confidence:',match_response['FaceMatches'][0]['Face']['Confidence'])
            else:
                print('No faces matched')
        except:
            print('No face detected')
    time.sleep(10)
"facial_matching.py" [dos] 40L, 1495B written                                                                    13,0-1        All
```

4. Test and Run

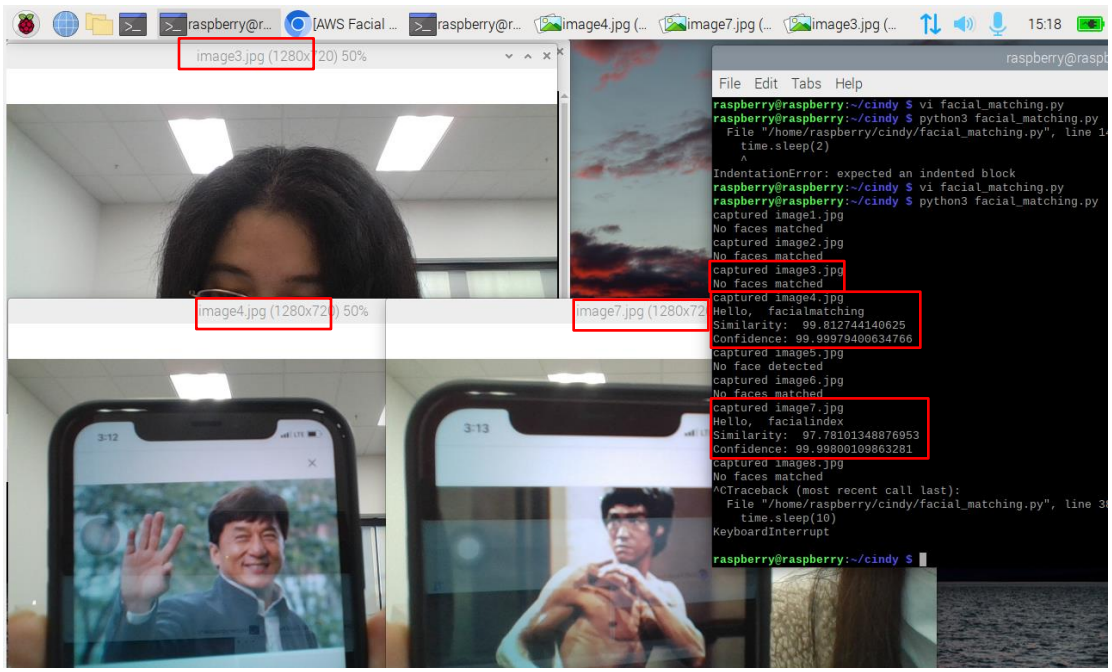
First, run facial_indexing.py first

$ python3 facial_indexing.py

```
raspberry@raspberry:~/cindy $ vi facial_indexing.py
raspberry@raspberry:~/cindy $ python3 facial_indexing.py
indexing:  bruceleeimages
FaceId:  613b82c2-7c66-4a3d-9a62-070dac27893f
indexing:  bruceleeimages
FaceId:  5318211e-9ca4-4e38-97cd-09bb50fbd16f
indexing:  bruceleeimages
FaceId:  96a8a99e-ea38-4876-b365-33ff98df0dbe
indexing:  jackiechanimages
FaceId:  3fc80101-4f20-4703-a449-3bfd5bcf6c59
indexing:  jackiechanimages
FaceId:  2989990d-b8fb-4f8a-945c-13dbe2c06ada
raspberry@raspberry:~/cindy $
```

Second, run facial_matching.py

$ python3 facial_matching.py



Done!!!

Reference Link:

[4.6. Webcam Passthrough (oracle.com)]

[Enabling Webcam in VirtualBox Guest OS on Windows Host (scribles.net)]