Week 12: Homework 1: Project: Facial Recognition on Raspberry Pi with AWS Rekognition

https://hc.labnet.sfbu.edu/~henry/npu/classes//iot/learning_aws_iot/slide/exercise_learning_aws_iot.html

Q8 ==> Project: Facial Recognition on Raspberry Pi with AWS Rekognition

1. Project: Facial Recognition on Raspberry Pi with AWS Rekognition
   - Process
     - Step 1: Prepare
       - Raspberry Pi emulator + VirtualBox
       - Integration of WebCam with Raspberry Pi
         - Raspberry PI Desktop with Webcam by Professor Adam Weng: Raspberry PI Desktop (i.e., Raspberry Pi Emulator) + PC Webcam
     - Step 2: Continue the proces of Facial Recognition on Raspberry Pi with AWS Rekognition

       - Hint
         - The Python code needs to be modified for the Integration of WebCam with Raspberry Pi
       - References

         - Facial Recognition on Raspberry Pi with AWS Rekognition - Youtube

           - Data Analytics on Amazon Web Service
             - Introduce basic usage of Amazon Web Service (AWS) for data analysis and visualization, including python programming, data mining, data visualization and machine learning.
     - Step 3: Update your portfolio about this project
     - Step 4: Submit a PDF file document showing the procedure as part of the homework answers.
       Step 5: Submit the URL of your GitHub webpage as part of the homework answers.
       GitHub directory structure

       IoT

       ```
           AWS IoT + Raspberry Pi Emulator + Image
       and Video Analysis
       ```
       References

2022 Fall

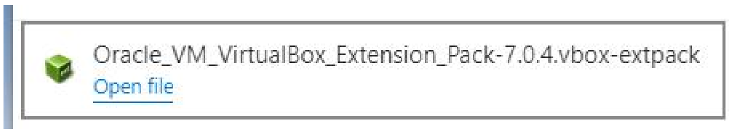
Step 1: Test Laptop Camera with Raspberry Pi Desktop on VirtualBox

1) Install the VirtualBox Extension Pack

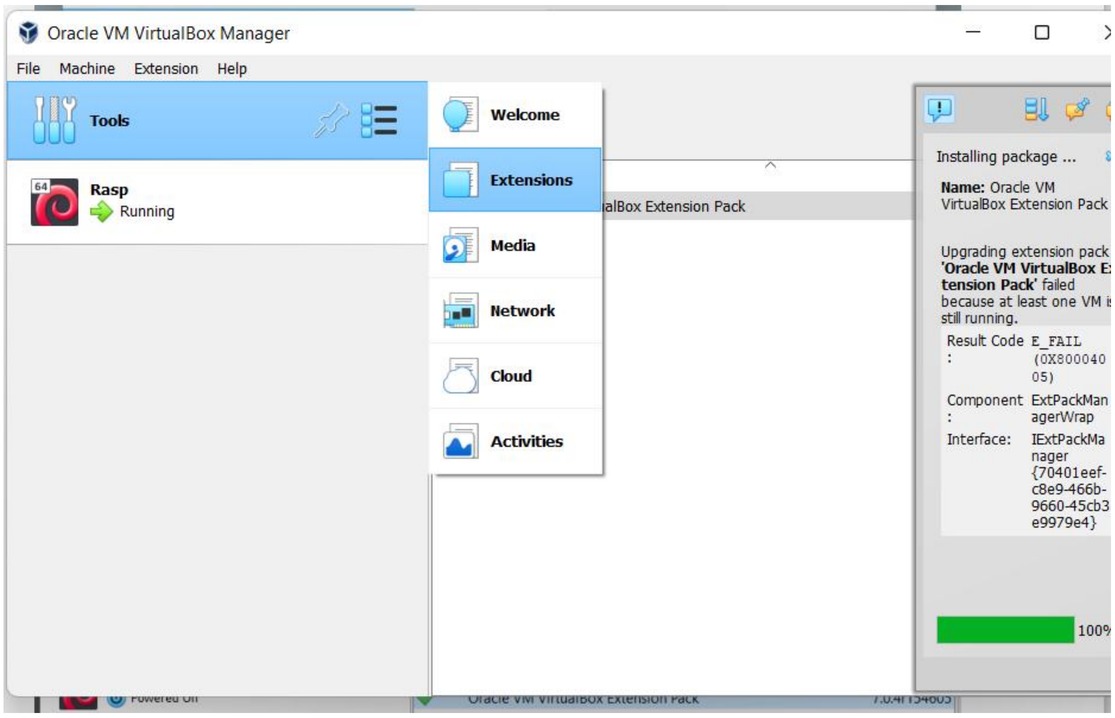
Downloads – Oracle VM VirtualBox

VirtualBox 7.0.4 Oracle VM VirtualBox Extension Pack

- All supported platforms


Oracle_VM_VirtualBox_Extension_Pack-7.0.4.vbox-extpack
Open file

After downloading, open it for automatically install
Check Installation

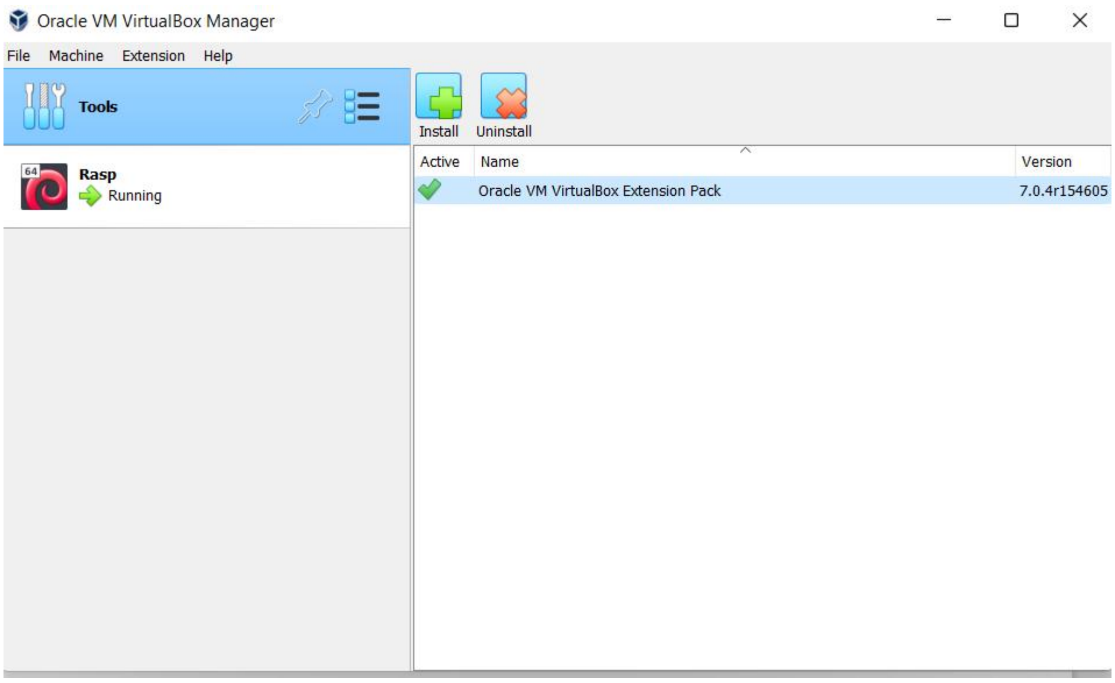


Installing package ...

Name: Oracle VM VirtualBox Extension Pack

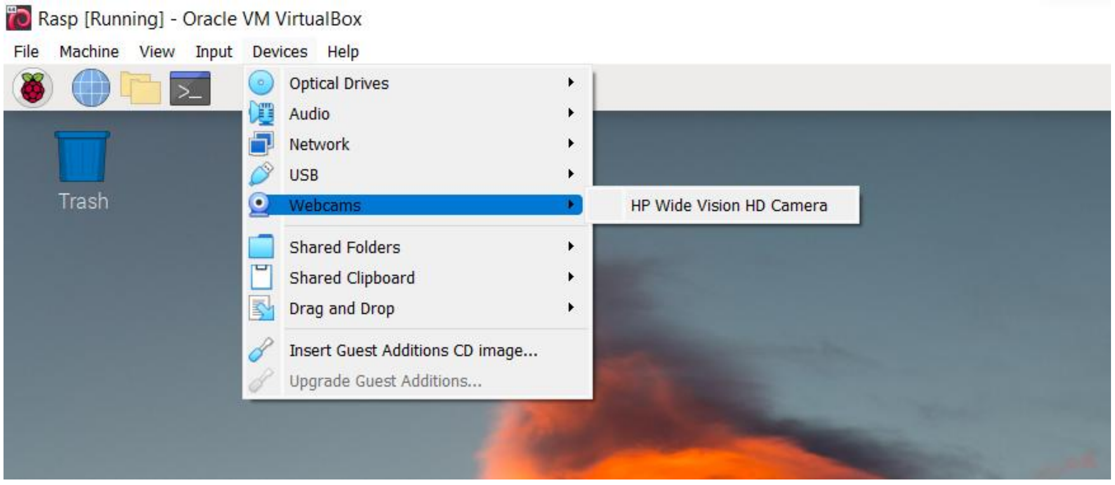
Upgrading extension pack **'Oracle VM VirtualBox Extension Pack'** failed because at least one VM is still running.

Result Code: E_FAIL (0X80004005)
Component: ExtPackManagerWrap
Interface: IExtPackManager {70401eef-c8e9-466b-9660-45cb3e9979e4}



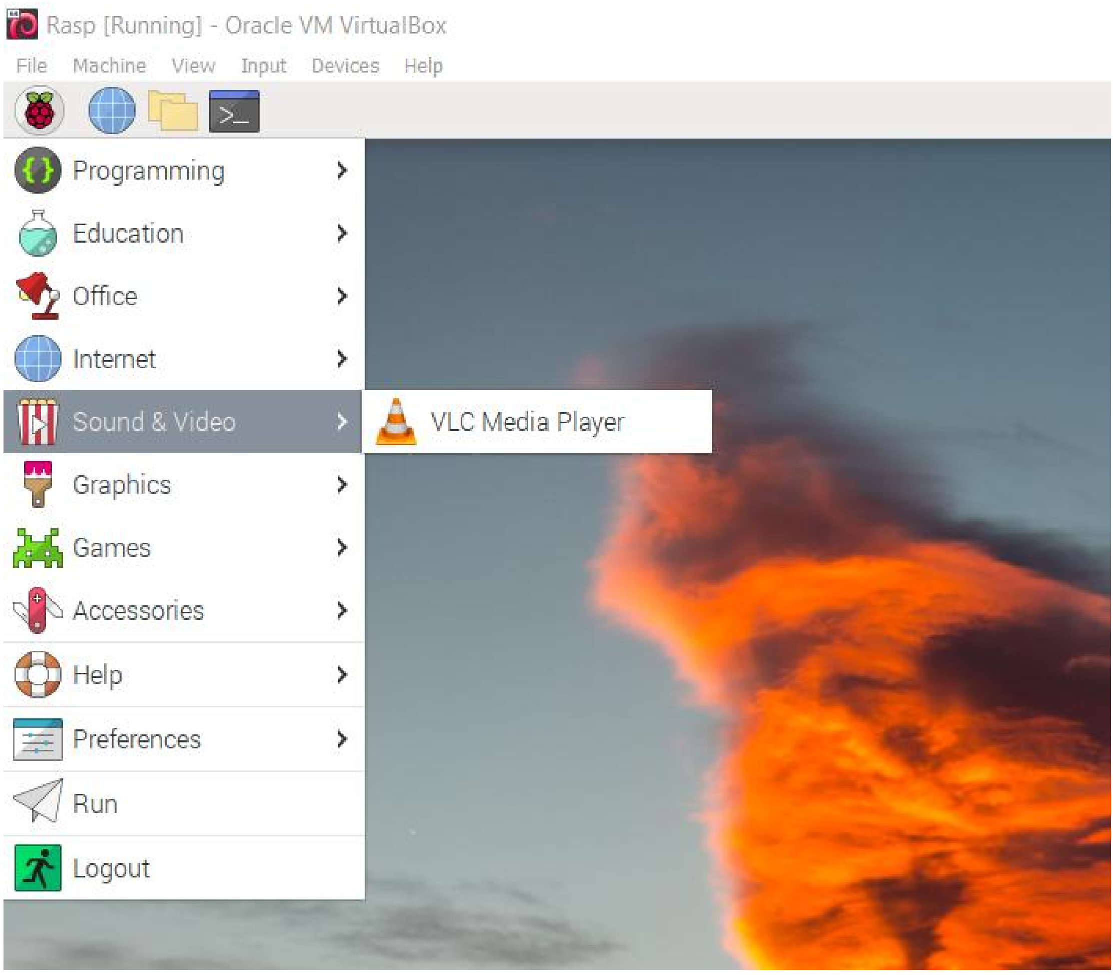
Active	Name	Version
✓	Oracle VM VirtualBox Extension Pack	7.0.4r154605

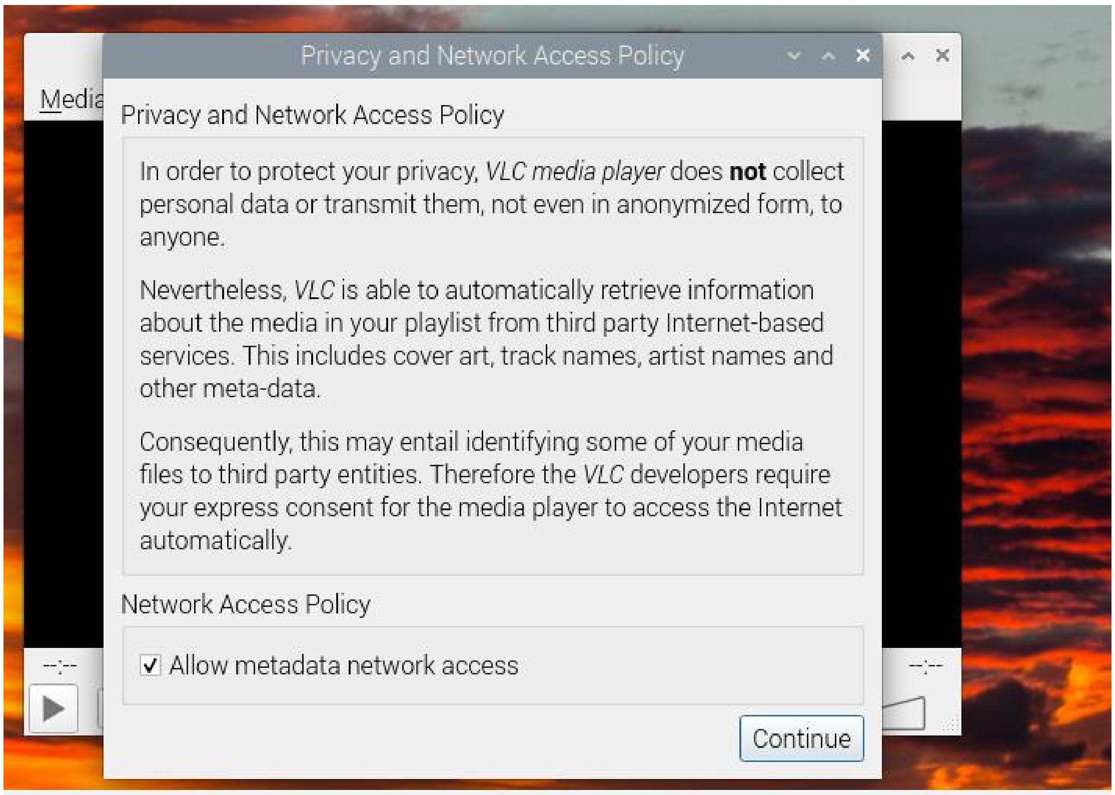
2) Open Raspberry Pi Desktop and enable Camera



3) Test Camera with VLC Media Player

➔ Open VLC Media Player





Privacy and Network Access Policy

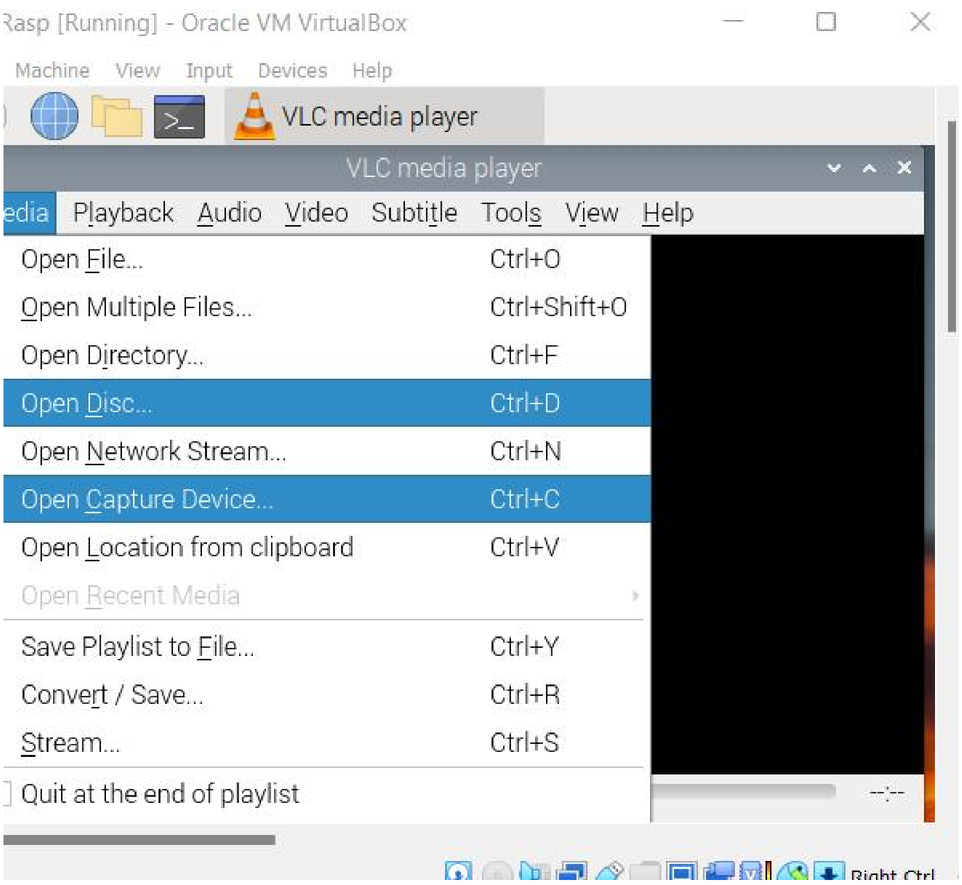
In order to protect your privacy, *VLC media player* does **not** collect personal data or transmit them, not even in anonymized form, to anyone.

Nevertheless, *VLC* is able to automatically retrieve information about the media in your playlist from third party Internet-based services. This includes cover art, track names, artist names and other meta-data.

Consequently, this may entail identifying some of your media files to third party entities. Therefore the *VLC* developers require your express consent for the media player to access the Internet automatically.

Network Access Policy

✓ Allow metadata network access



Rasp [Running] - Oracle VM VirtualBox

Machine View Input Devices Help

VLC media player

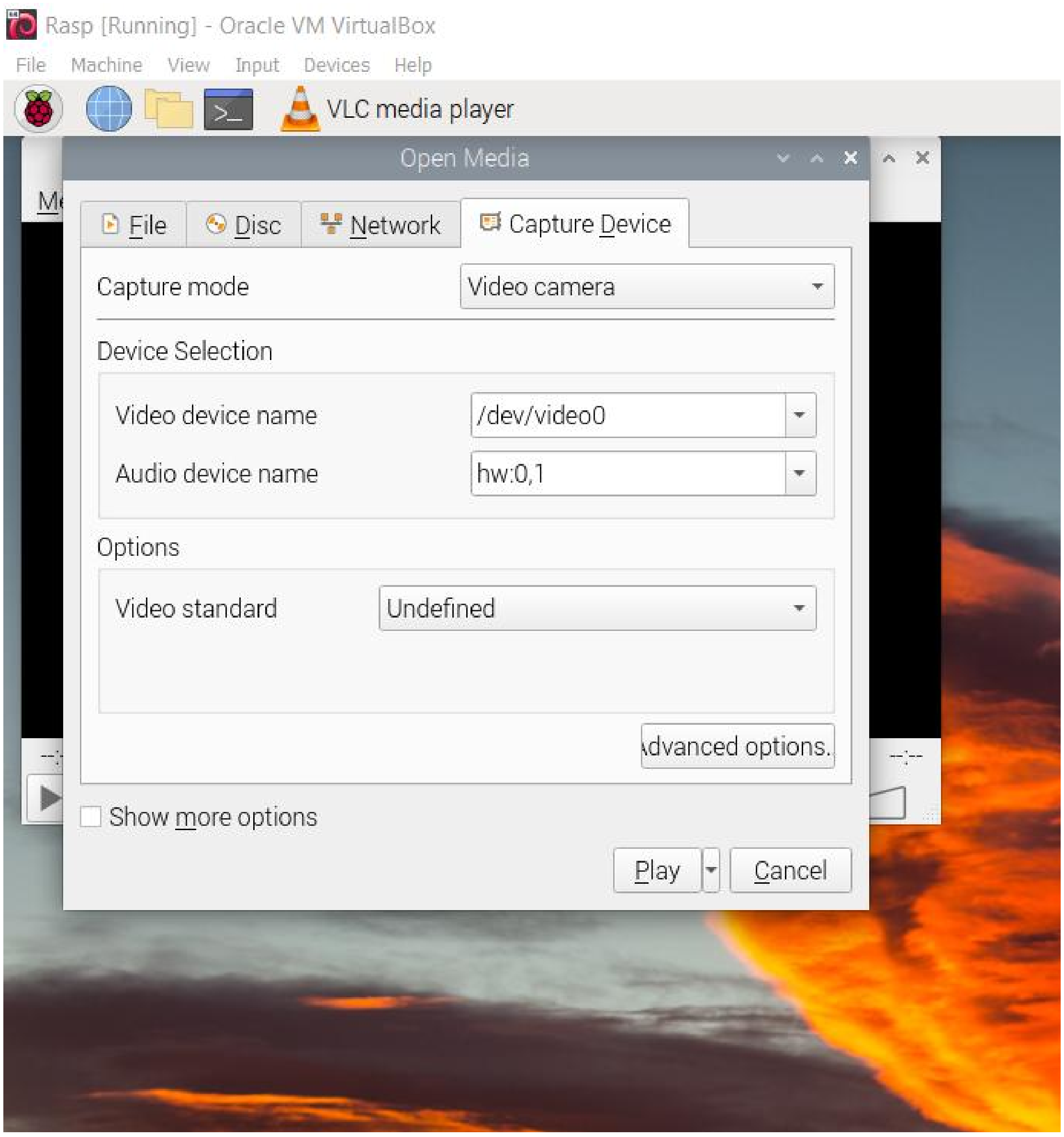
edia Playback Audio Video Subtitle Tools View Help

Open File...	Ctrl+O
Open Multiple Files...	Ctrl+Shift+O
Open Directory...	Ctrl+F
Open Disc...	Ctrl+D
Open Network Stream...	Ctrl+N
Open Capture Device...	Ctrl+C
Open Location from clipboard	Ctrl+V
Open Recent Media	
Save Playlist to File...	Ctrl+Y
Convert / Save...	Ctrl+R
Stream...	Ctrl+S
Quit at the end of playlist	

Right Ctrl

➔ Type name /dev/video0, then click Play

➔ Start capture


Rasp [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

VLC media player

Open Media

File | Disc | Network | Capture Device

Capture mode: Video camera

Device Selection

Video device name: /dev/video0

Audio device name: hw:0,1

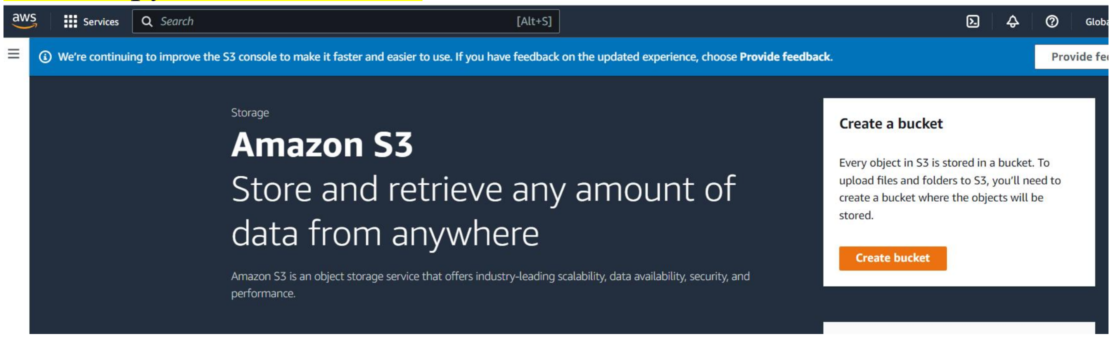
Options

Video standard: Undefined

Step 2: Facial Recognition on Raspberry Pi with AWS Rekognition

1) AWS Rekognition setup

Note: copy the bucket name



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Storage

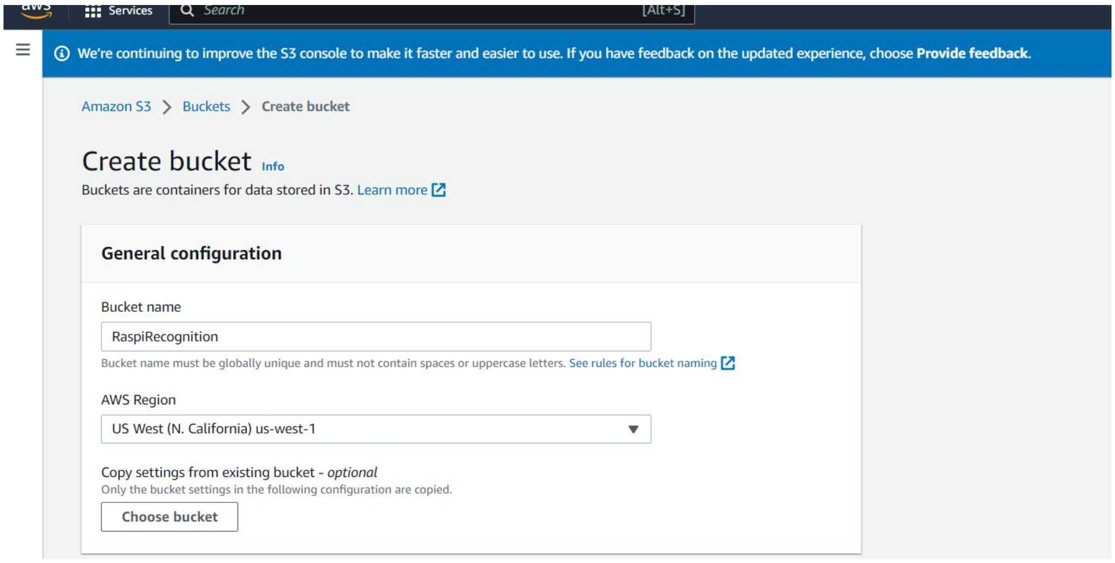
Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

RaspiRecognition

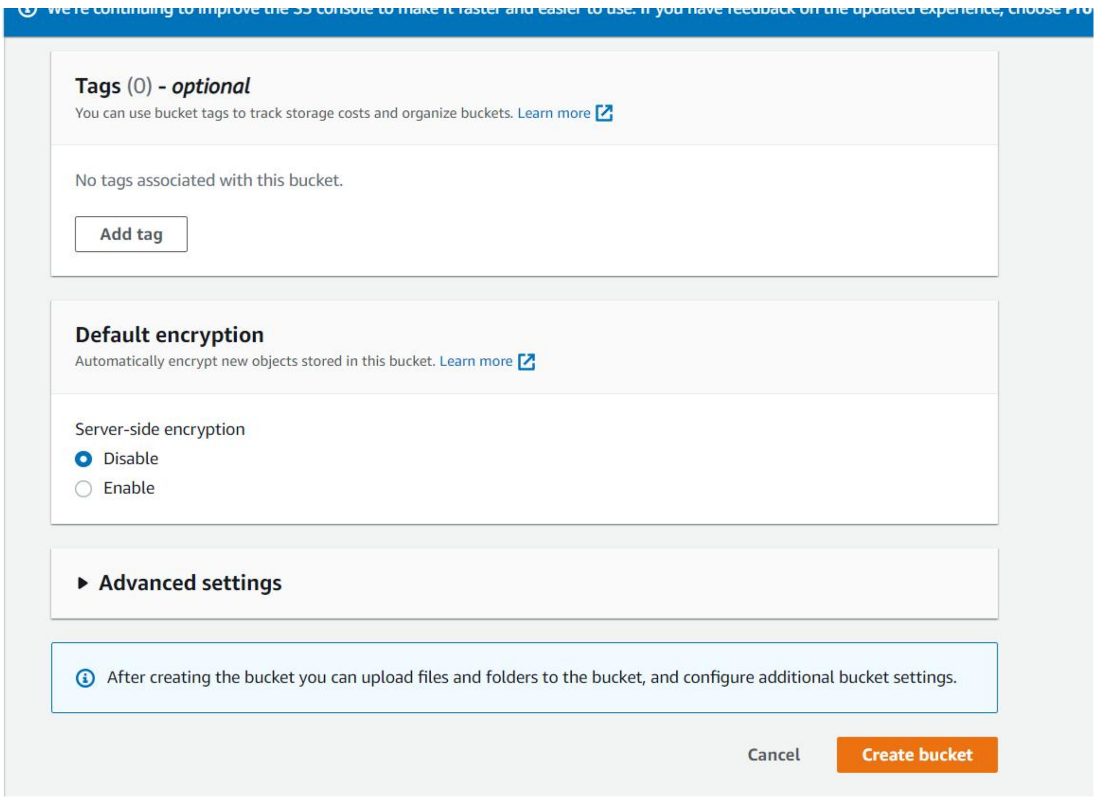
Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US West (N. California) us-west-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.



Tags (0) - ***optional***

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Default encryption

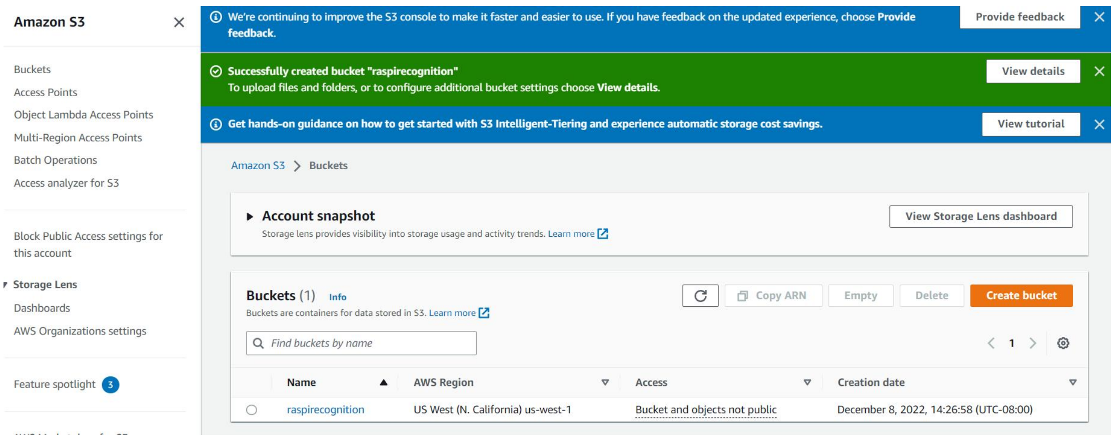
Automatically encrypt new objects stored in this bucket. Learn more

Server-side encryption

[✓ Disable] Enable

Advanced settings

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.


Amazon S3

Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3

Block Public Access settings for this account

Storage Lens
Dashboards
AWS Organizations settings

Feature spotlight 3

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully created bucket "raspirecognition"
To upload files and folders, or to configure additional bucket settings choose **View details**.

Get hands-on guidance on how to get started with S3 Intelligent-Tiering and experience automatic storage cost savings.

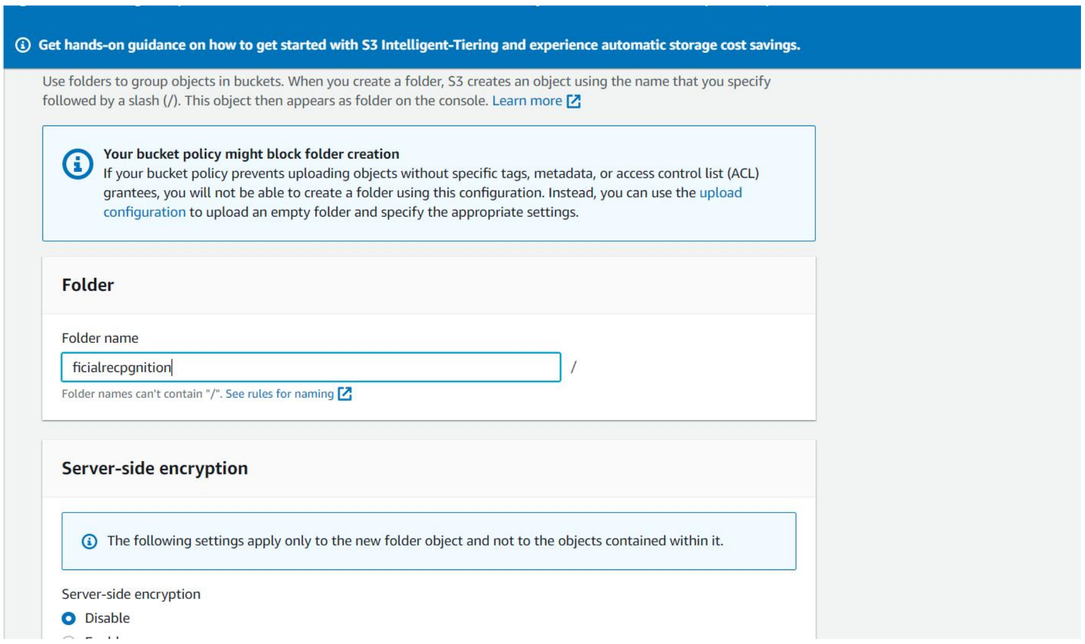
Amazon S3 > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	raspirecognition	US West (N. California) us-west-1	Bucket and objects not public	December 8, 2022, 14:26:58 (UTC-08:00)

Upload 5 photos:
Create folder:
Copy the folder name


Get hands-on guidance on how to get started with S3 Intelligent-Tiering and experience automatic storage cost savings.

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. Learn more

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.

Folder

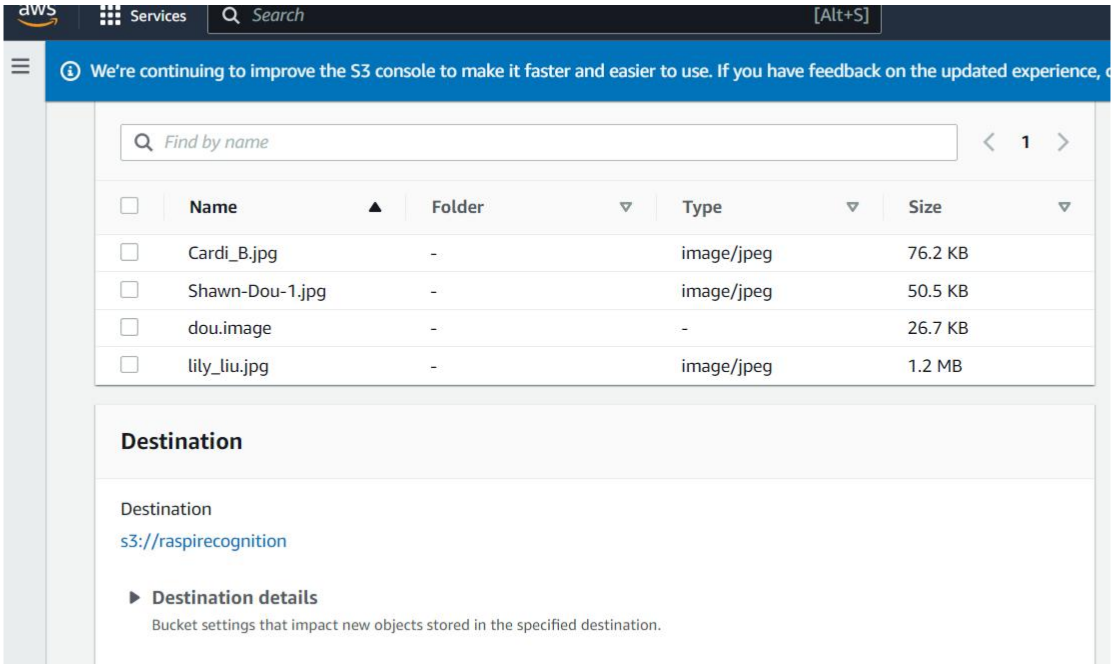
Folder name
ficialrecpgnition /
Folder names can't contain "/". See rules for naming

Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

Server-side encryption
[✓ Disable]

Upload 5 photos:

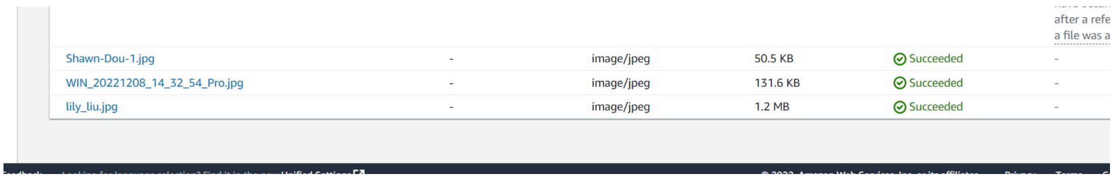

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience,

Name	Folder	Type	Size
Cardi_B.jpg	-	image/jpeg	76.2 KB
Shawn-Dou-1.jpg	-	image/jpeg	50.5 KB
dou.image	-	-	26.7 KB
lily_liu.jpg	-	image/jpeg	1.2 MB

Destination

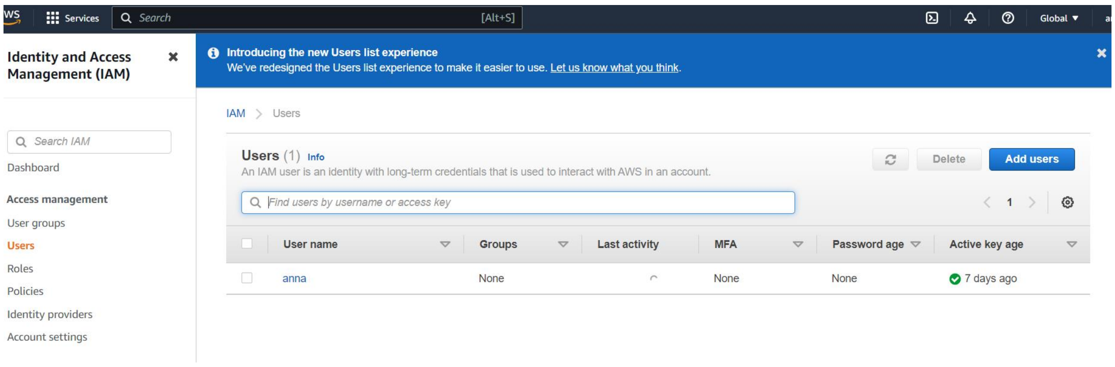
Destination
s3://raspirecognition

Destination details
Bucket settings that impact new objects stored in the specified destination.


after a refe
a file was a

Shawn-Dou-1.jpg	-	image/jpeg	50.5 KB	Succeeded	-
WIN_20221208_14_32_54_Pro.jpg	-	image/jpeg	131.6 KB	Succeeded	-
lily_liu.jpg	-	image/jpeg	1.2 MB	Succeeded	-

➔ Go to IAM to Add Permissions

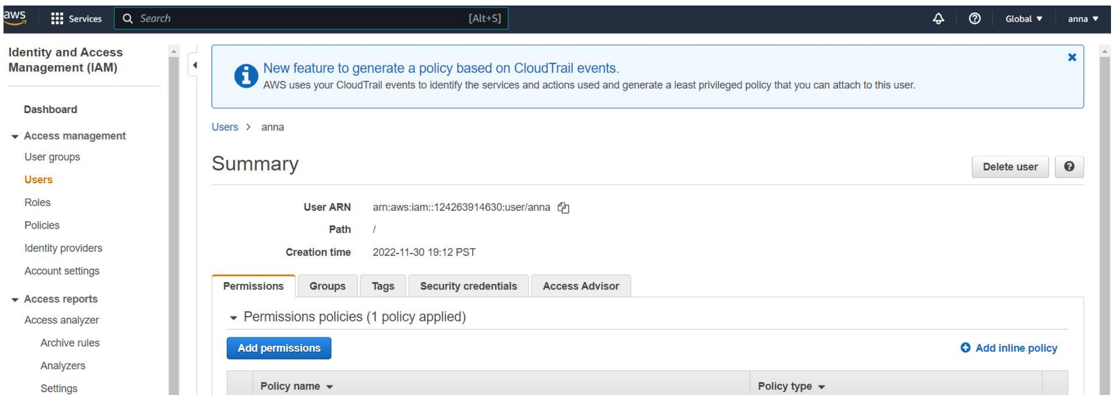

Identity and Access Management (IAM)

Introducing the new Users list experience
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

IAM > Users

Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active key age
anna	None		None	None	7 days ago


Identity and Access Management (IAM)

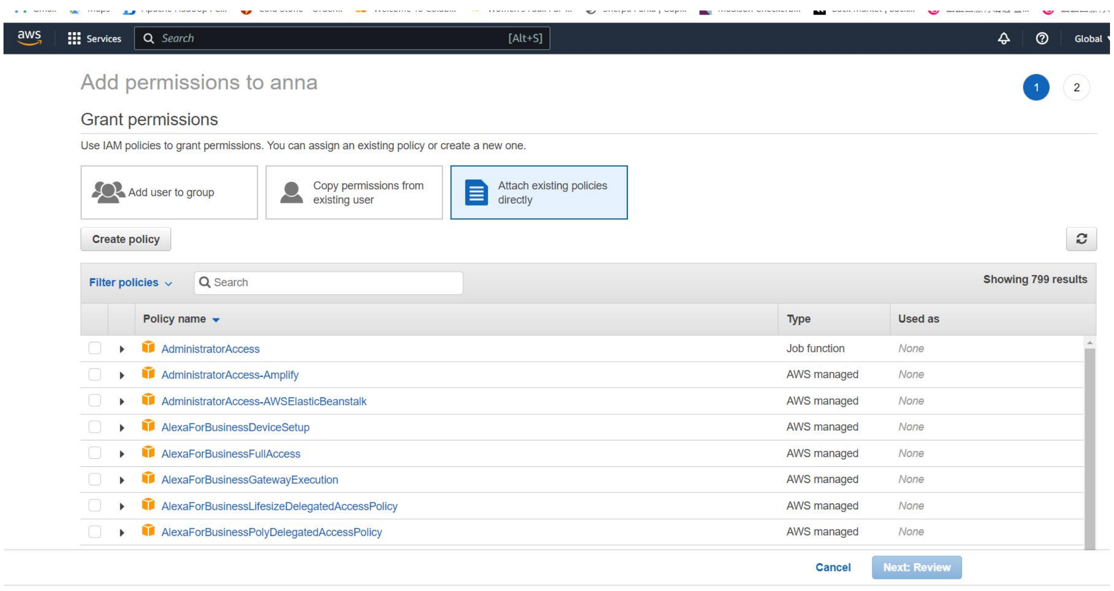
New feature to generate a policy based on CloudTrail events.
AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this user.

Users > anna

Summary

User ARN	arn:aws:iam::124263914630:user/anna
Path	/
Creation time	2022-11-30 19:12 PST

Permissions policies (1 policy applied)



Add permissions to anna

1 2

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

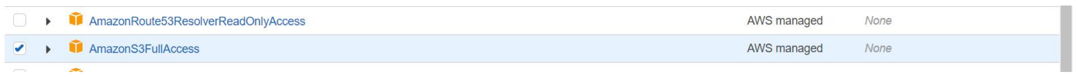
Add user to group | Copy permissions from existing user | Attach existing policies directly

Create policy

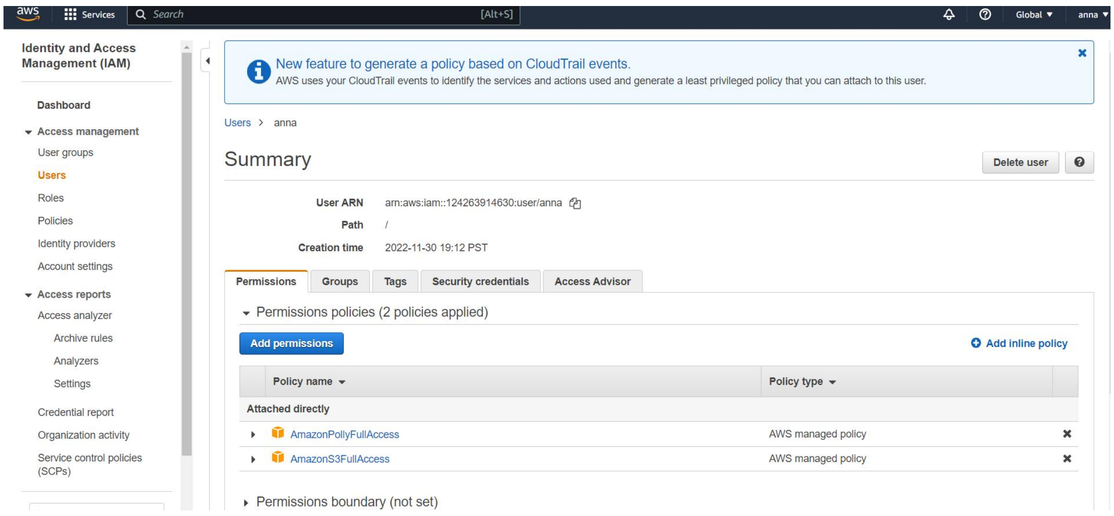
Filter policies | Search | Showing 799 results

Policy name	Type	Used as
AdministratorAccess	Job function	None
AdministratorAccess-Amplify	AWS managed	None
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
AlexaForBusinessDeviceSetup	AWS managed	None
AlexaForBusinessFullAccess	AWS managed	None
AlexaForBusinessGatewayExecution	AWS managed	None
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None

Cancel | Next: Review



	Policy name	Type	Used as
☐	AmazonRoute53ResolverReadOnlyAccess	AWS managed	None
✓	AmazonS3FullAccess	AWS managed	None



New feature to generate a policy based on CloudTrail events.
AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this user.

Users > anna

Summary

Delete user

User ARN arn:aws:iam::124263914630:user/anna
Path /
Creation time 2022-11-30 19:12 PST

Permissions | Groups | Tags | Security credentials | Access Advisor

Permissions policies (2 policies applied)

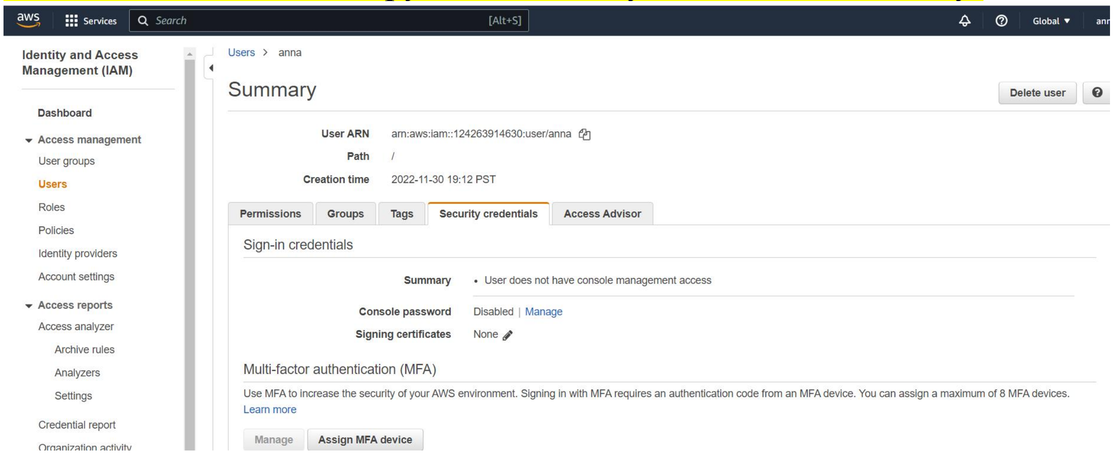
Add permissions | Add inline policy

Policy name	Policy type
Attached directly	
AmazonPollyFullAccess	AWS managed policy
AmazonS3FullAccess	AWS managed policy

Permissions boundary (not set)

Access Keys are under Security credentials

Download cvs.file and copy the access key and access secret key:



Users > anna

Summary

Delete user

User ARN arn:aws:iam::124263914630:user/anna
Path /
Creation time 2022-11-30 19:12 PST

Permissions | Groups | Tags | Security credentials | Access Advisor

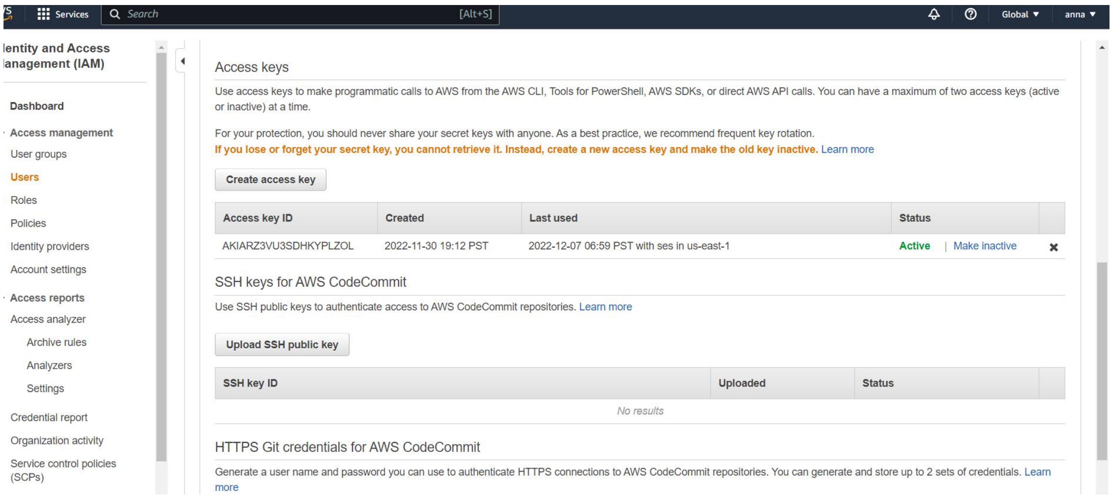
Sign-in credentials

Summary • User does not have console management access
Console password Disabled | Manage
Signing certificates None

Multi-factor authentication (MFA)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. You can assign a maximum of 8 MFA devices. Learn more

Manage | Assign MFA device



Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more

Create access key

Access key ID	Created	Last used	Status
AKIARZ3VU3SDHKYPLZOL	2022-11-30 19:12 PST	2022-12-07 06:59 PST with ses in us-east-1	Active | Make inactive

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

Upload SSH public key

SSH key ID	Uploaded	Status
No results		

HTTPS Git credentials for AWS CodeCommit

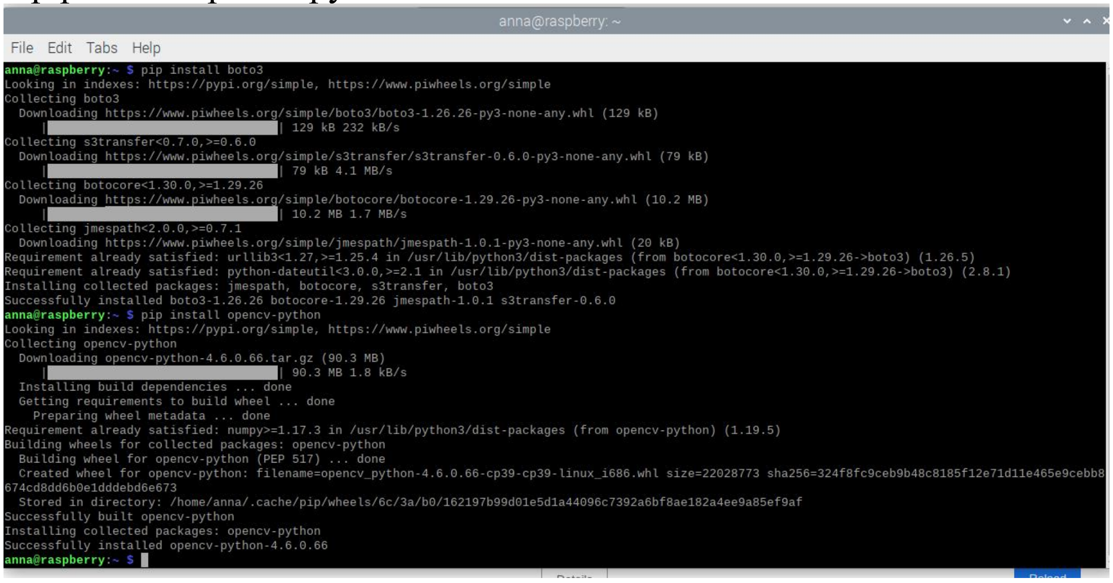
Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate and store up to 2 sets of credentials. Learn more

Step 3.Test

a) Prepare code:

$ pip install boto3

$ pip install opencv-python

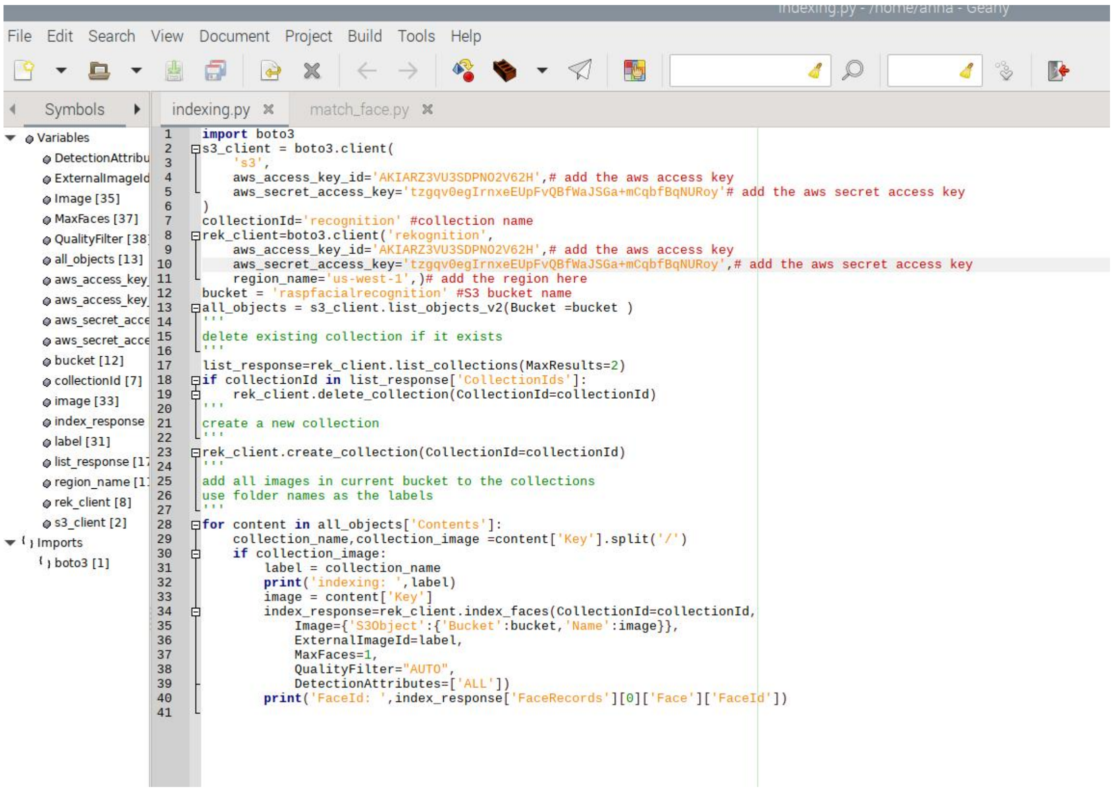


```
anna@raspberry:~ $ pip install boto3
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting boto3
  Downloading https://www.piwheels.org/simple/boto3/boto3-1.26.26-py3-none-any.whl (129 kB)
     |████████████████████████████████| 129 kB 232 kB/s
Collecting s3transfer<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 4.1 MB/s
Collecting botocore<1.30.0,>=1.29.26
  Downloading https://www.piwheels.org/simple/botocore/botocore-1.29.26-py3-none-any.whl (10.2 MB)
     |████████████████████████████████| 10.2 MB 1.7 MB/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)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3/dist-packages (from botocore<1.30.0,>=1.29.26->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,>=1.29.26->boto3) (2.8.1)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.26.26 botocore-1.29.26 jmespath-1.0.1 s3transfer-0.6.0
anna@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 1.8 kB/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=22028773 sha256=324f8fc9ceb9b48c8185f12e71d11e465e9cebb8
674cd8dd6b0e1dddebd6e673
  Stored in directory: /home/anna/.cache/pip/wheels/6c/3a/b0/162197b99d01e5d1a44096c7392a6bf8ae182a4ee9a85ef9af
Successfully built opencv-python
Installing collected packages: opencv-python
Successfully installed opencv-python-4.6.0.66
anna@raspberry:~ $ 
```

Prepare code:

Remember to modify the codes with the previous bucket name, folder name, access key and access secret key in below codes:

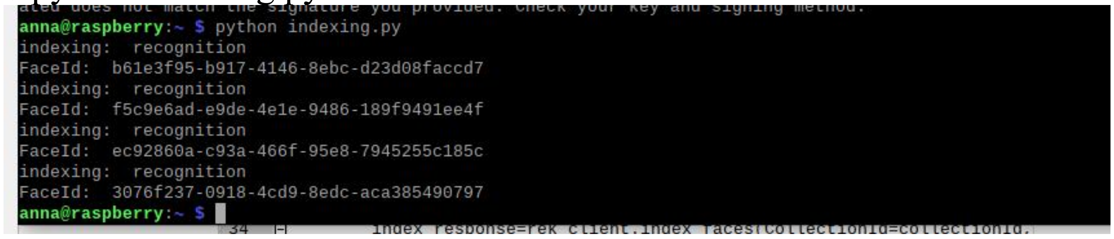
$ indexing.py



```
import boto3
s3_client = boto3.client(
    's3',
    aws_access_key_id='AKIARZ3VU3SDPNO2V62H',# add the aws access key
    aws_secret_access_key='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'# add the aws secret access key
)
collectionId='recognition' #collection name
rek_client=boto3.client('rekognition',
    aws_access_key_id='AKIARZ3VU3SDPNO2V62H',# 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 = 'raspfacialrecognition' #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'])

```

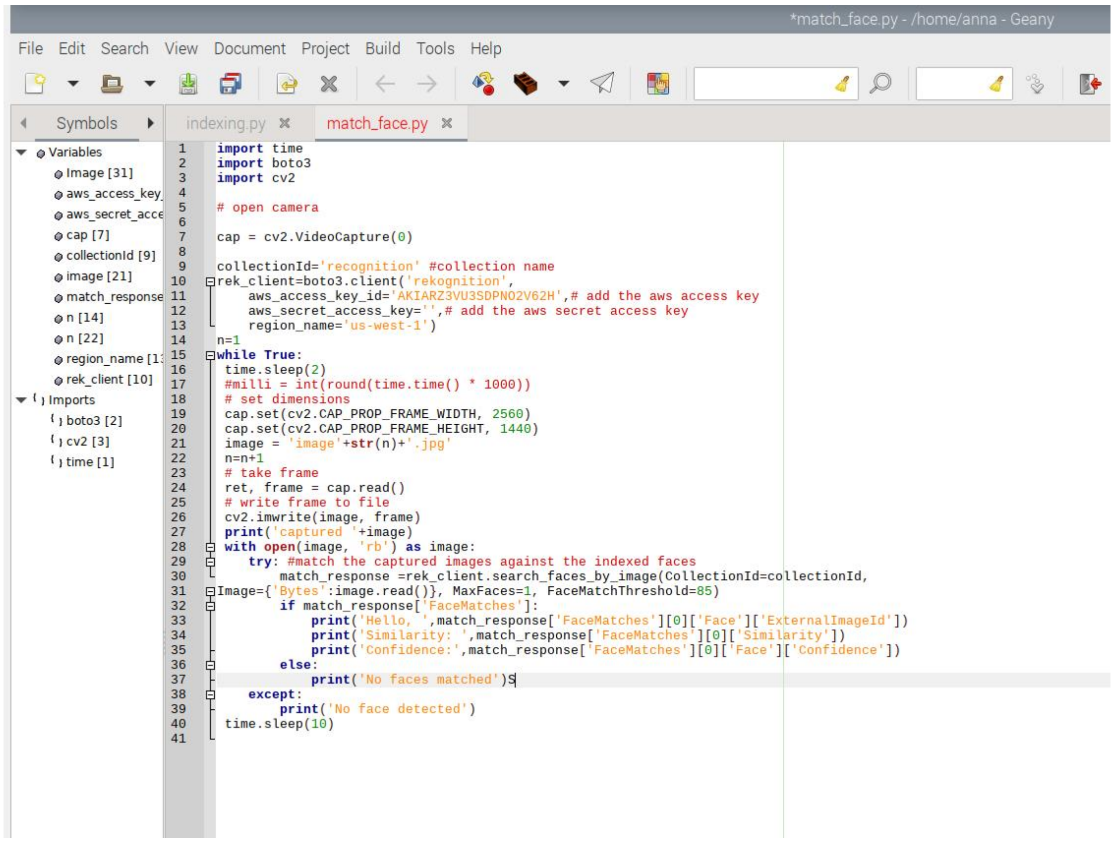
Run the code:

$ python indexing.py



```
ated does not match the signature you provided. Check your key and signing method.
anna@raspberry:~ $ python indexing.py
indexing:  recognition
FaceId:  b61e3f95-b917-4146-8ebc-d23d08faccd7
indexing:  recognition
FaceId:  f5c9e6ad-e9de-4e1e-9486-189f9491ee4f
indexing:  recognition
FaceId:  ec92860a-c93a-466f-95e8-7945255c185c
indexing:  recognition
FaceId:  3076f237-0918-4cd9-8edc-aca385490797
anna@raspberry:~ $ 
```

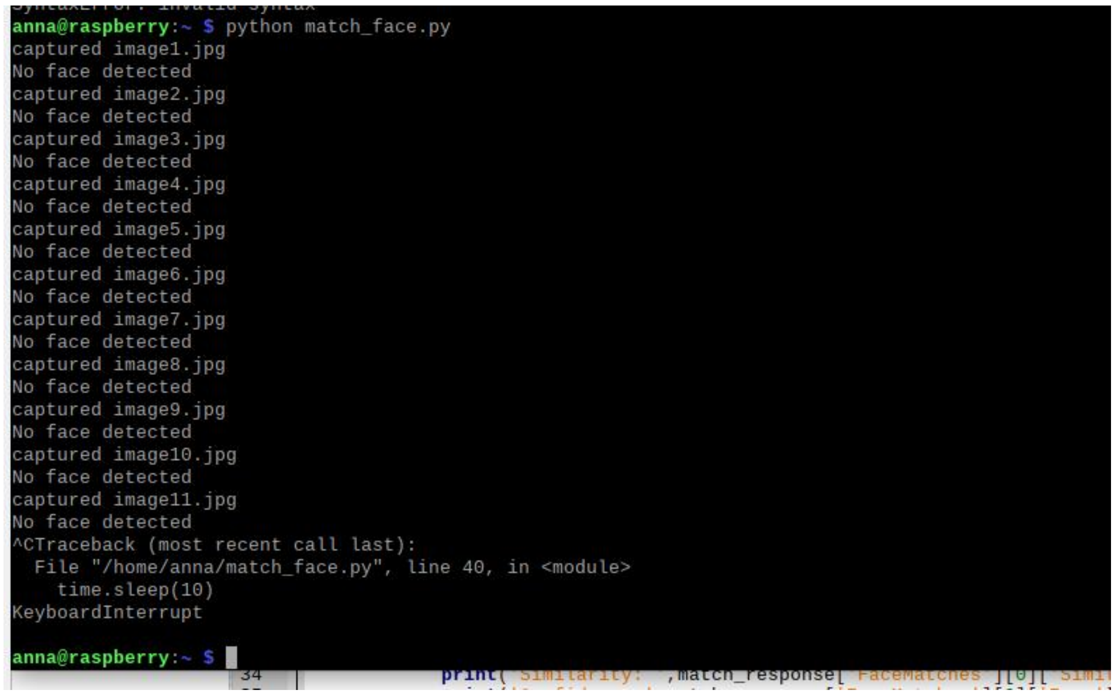
match_face.py



```python
import time
import boto3
import cv2

# open camera

cap = cv2.VideoCapture(0)

collectionId='recognition' #collection name
rek_client=boto3.client('rekognition',
    aws_access_key_id='AKIARZ3VU3SDPNO2V62H',# add the aws access key
    aws_secret_access_key='',# 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')S
    except:
        print('No face detected')
 time.sleep(10)
```

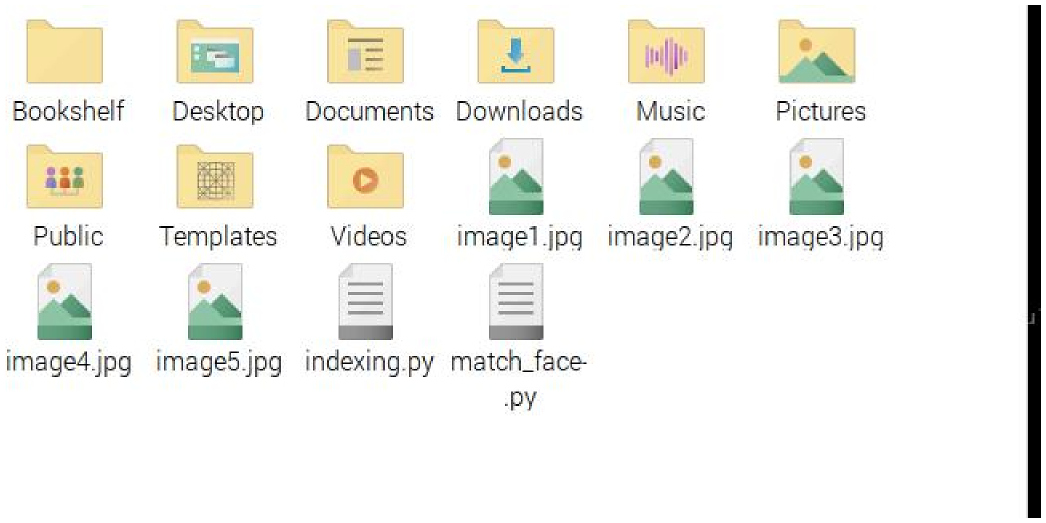
Run the code:
$ python match_face.py

Meanwhile, open the camera to recognition:



```
anna@raspberry:~ $ python match_face.py
captured image1.jpg
No face detected
captured image2.jpg
No face detected
captured image3.jpg
No face detected
captured image4.jpg
No face detected
captured image5.jpg
No face detected
captured image6.jpg
No face detected
captured image7.jpg
No face detected
captured image8.jpg
No face detected
captured image9.jpg
No face detected
captured image10.jpg
No face detected
captured image11.jpg
No face detected
^CTraceback (most recent call last):
  File "/home/anna/match_face.py", line 40, in <module>
    time.sleep(10)
KeyboardInterrupt

anna@raspberry:~ $
```

Result:


Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
image1.jpg
image2.jpg
image3.jpg
image4.jpg
image5.jpg
indexing.py
match_face-.py