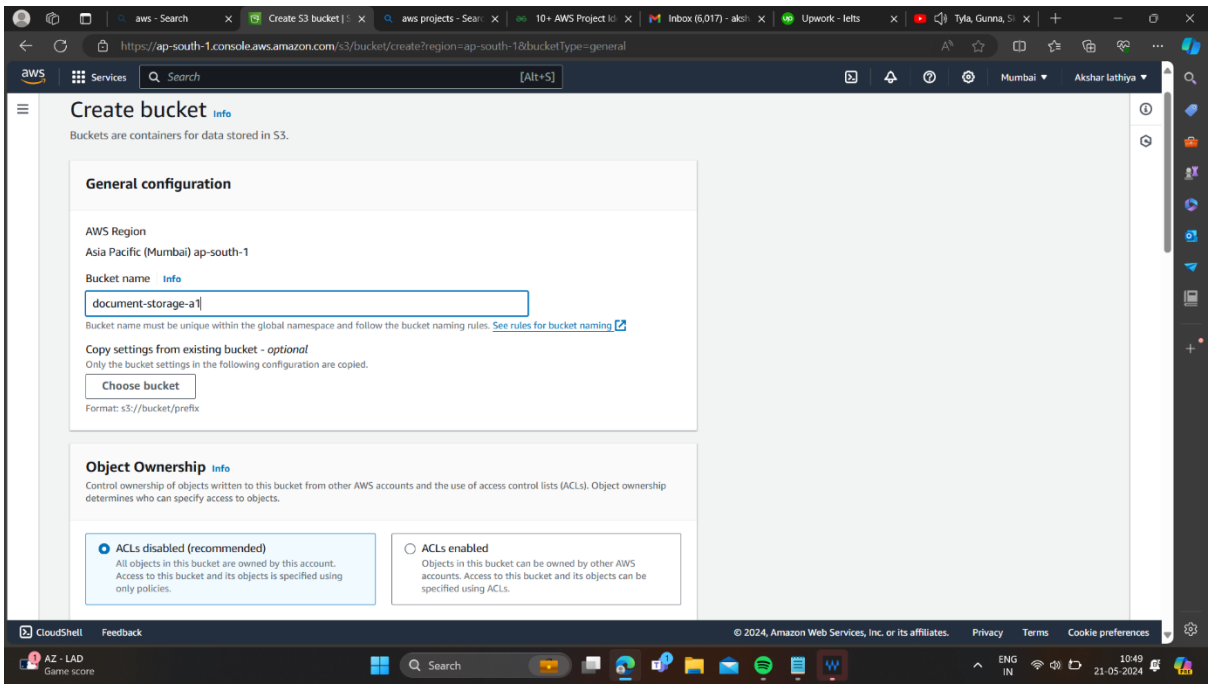
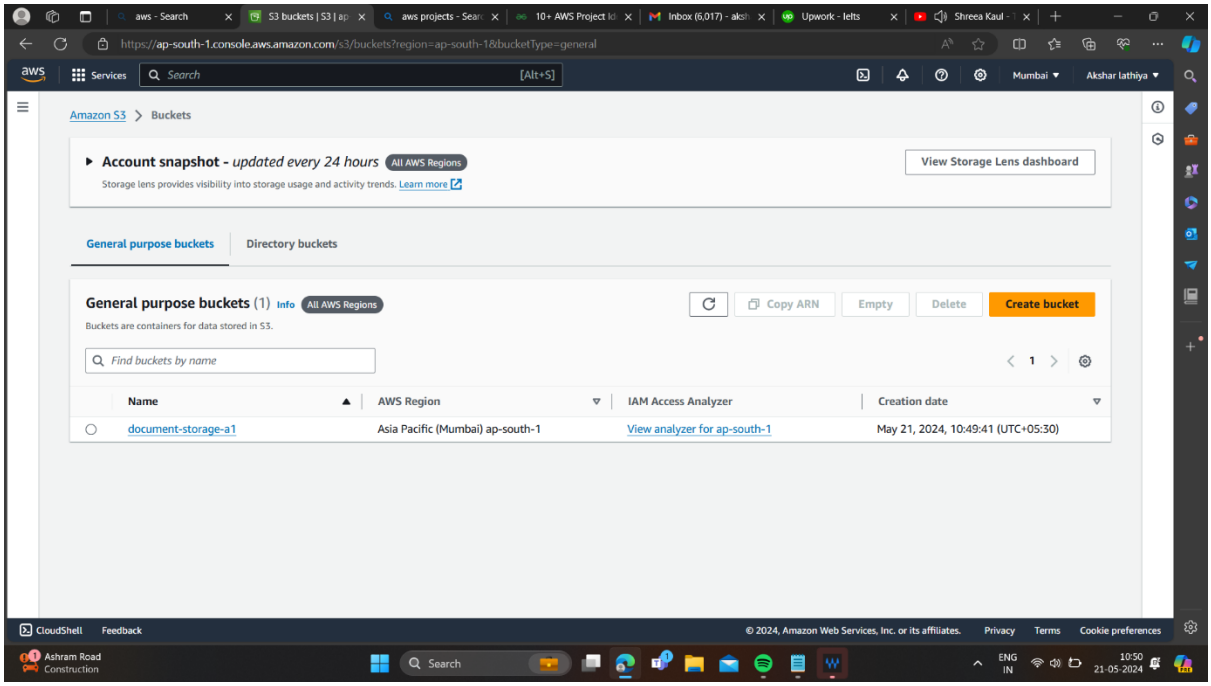
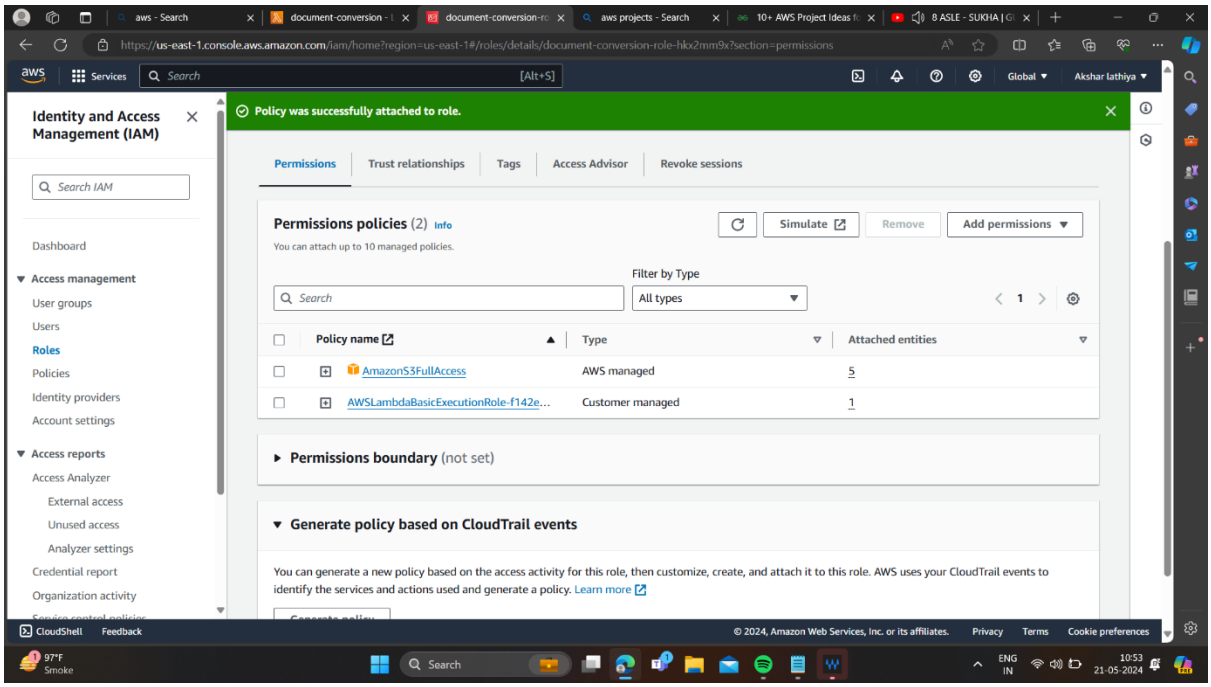
PDF conversion of any document uploaded in S3 bucket

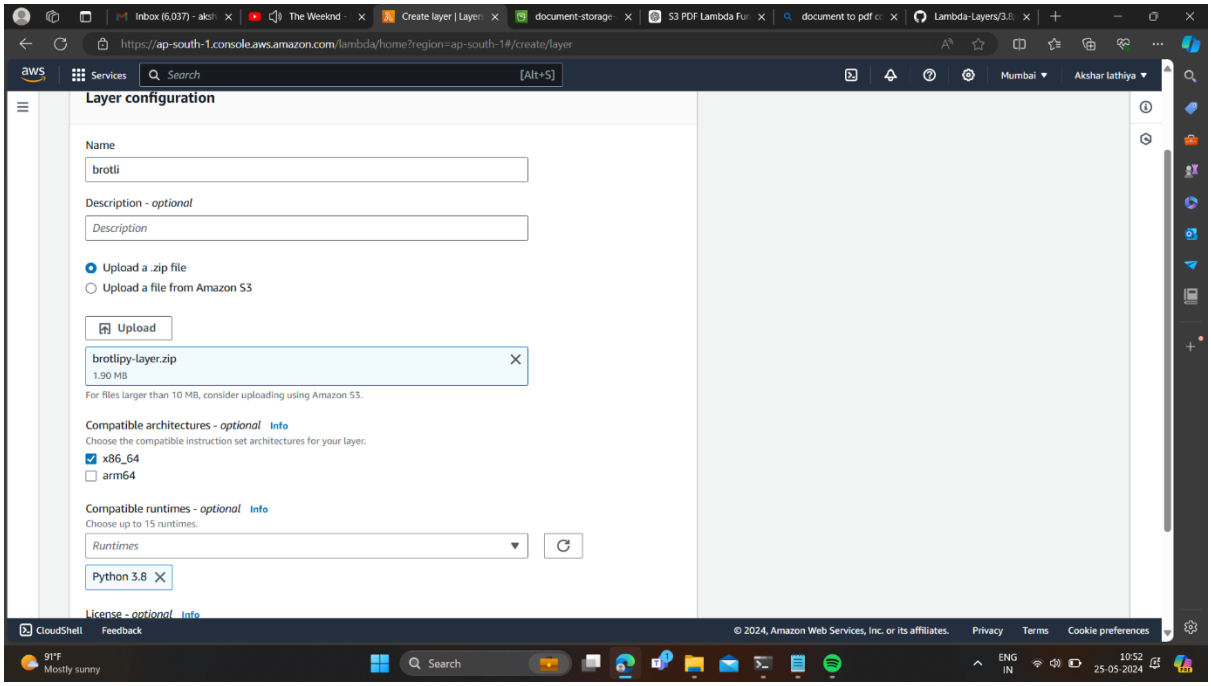
- **S3 Bucket setup:-**
  - Firstly , setup two buckets for source and destination.
  - Set both buckets as per default buckets configurations and make sure that both buckets are in same region.





- **Lambda Function setup:-**
  - Now configure lambda function and add python 3.8 as runtime .
  - The function contains layers , so add layer in section and configure by uploading brotli-layer.zip file to it.
  - Layers will add dependency required while running lambda function .
  - Add layer created in function and give aws arn of layer created .
  - Add aws arn of libreoffice of the Mumbai region.
  - Now insert python code for file conversion.
  - Configure memory and timeout for function.
  - Configure IAM role of lambda and give full S3 access to upload and access files in bucket.




Layer configuration
Name
brotli
Description - optional
Description
Upload a .zip file
Upload a file from Amazon S3
Upload
brotlipy-layer.zip
1.90 MB
For files larger than 10 MB, consider uploading using Amazon S3.
Compatible architectures - optional Info
Choose the compatible instruction set architectures for your layer.
x86_64
arm64
Compatible runtimes - optional Info
Choose up to 15 runtimes.
Runtimes
Python 3.8
License - optional Info
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

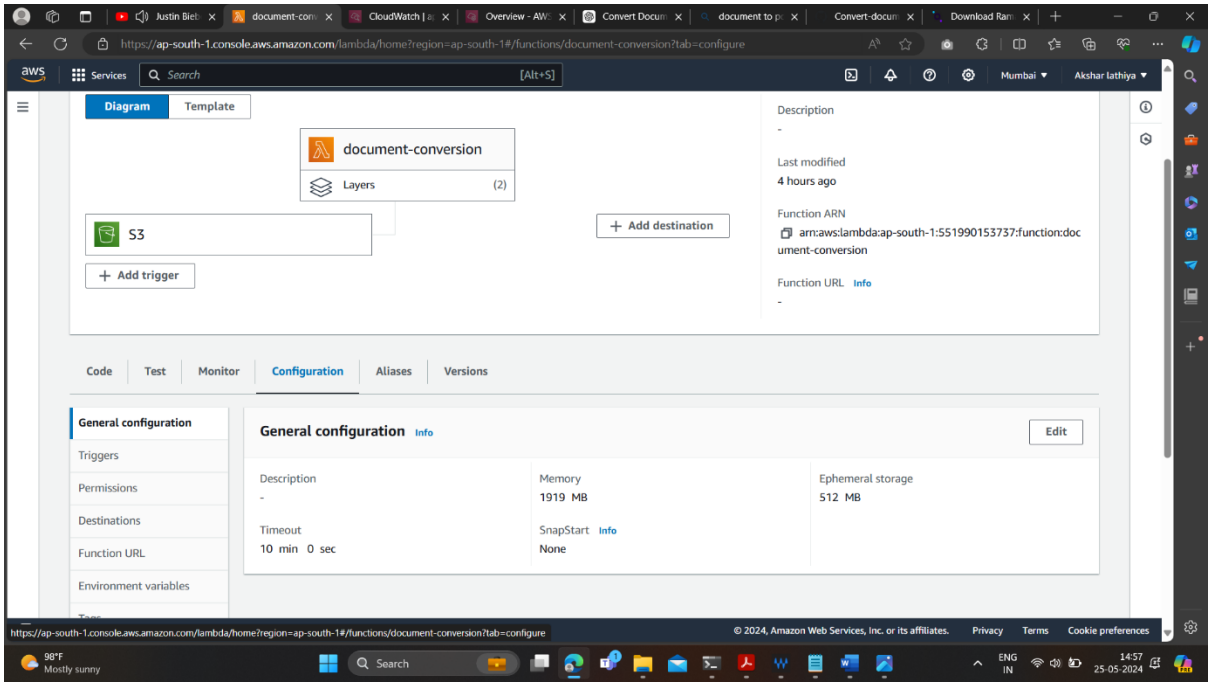
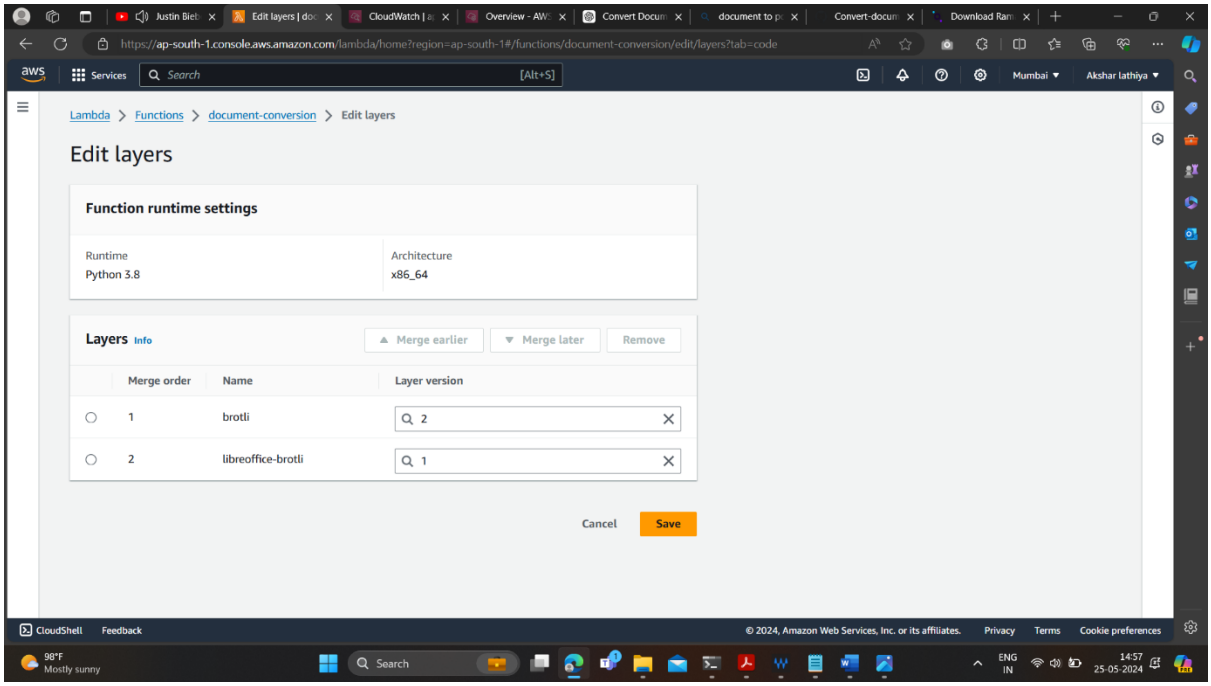
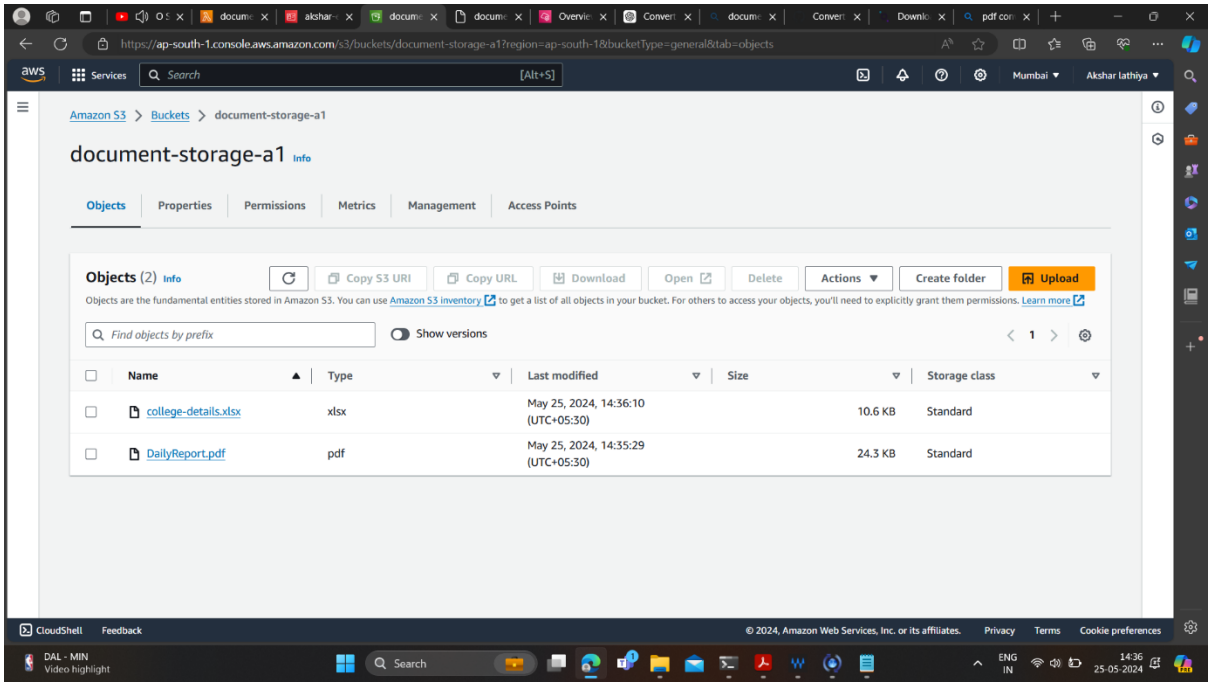
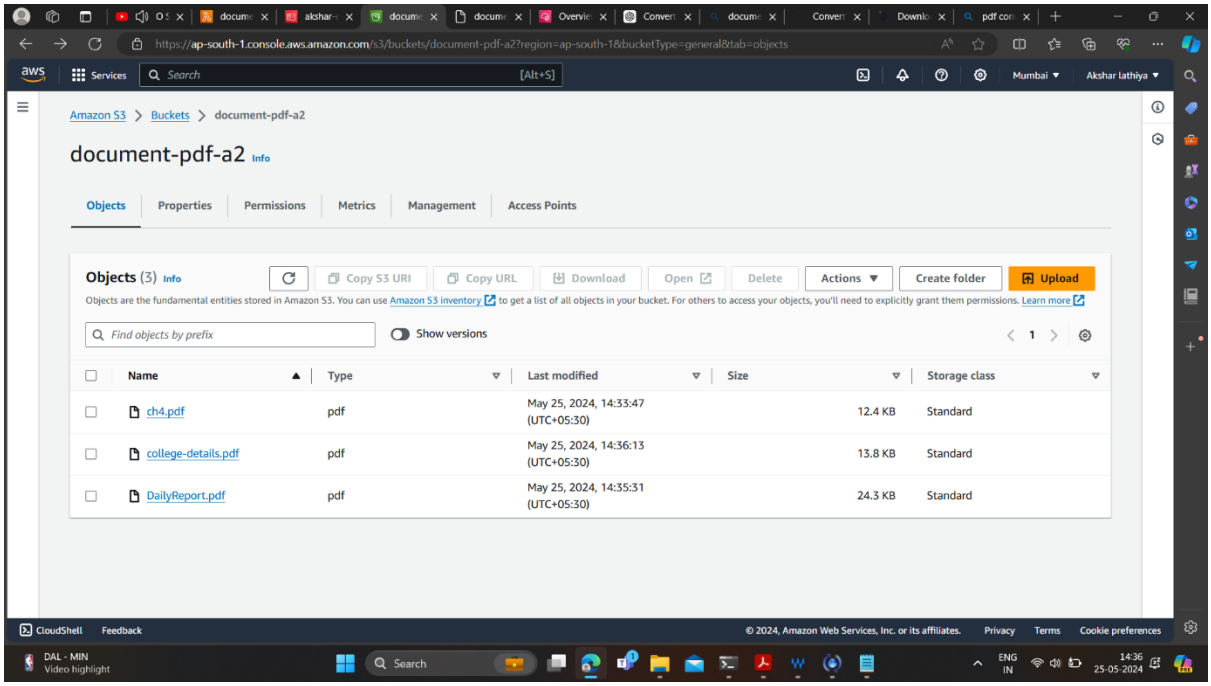


Diagram
Template
document-conversion
Layers
(2)
S3
Add trigger
Add destination
Description
-
Last modified
4 hours ago
Function ARN
arn:aws:lambda:ap-south-1:551990153737:function:document-conversion
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
General configuration Info
Edit
Description
-
Memory
1919 MB
Ephemeral storage
512 MB
Timeout
10 min 0 sec
SnapStart Info
None
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```python
import subprocess
import tarfile
import sys
from io import BytesIO
import os
import time
import boto3
import urllib.parse
import json

sys.path.append("/opt/brotli")
import brotli

ACCESS_KEY = "AKIAYBBJNKIEYBERKW4H"
SECRET_ACCESS_KEY = "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"

def extract_libre_office():
        buffer = BytesIO()
        with open('/opt/lo.tar.br', mode='rb') as fout:
            file = fout.read()
            buffer.write(brotli.decompress(file))
            buffer.seek(0)
            with tarfile.open(fileobj=buffer) as tar:
                tar.extractall('/tmp')

def lambda_handler(event, context):
    if os.path.exists("/tmp/instdir/program/soffice.bin"):
        pass
    else :
        # load libre
        extract_libre_office()

    # Get Trigger event
    if 'Records' in event.keys():
        input_bucket = event['Records'][0]['s3']['bucket']['name']
        output_bucket = "document-pdf-a2"
        input_key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key']
        s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY, aws_secret_access_key=SECR
    else :
        return 'test finished'

    # get S3 Object
    file_path = '/tmp/'+input_key
    s3.download_file(input_bucket, input_key, file_path)

    # Document -> pdf
    proc = subprocess.run("/tmp/instdir/program/soffice.bin --headless --norestore --inv
    print('STDOUT: {}'.format(proc.stdout))
    print('STDERR: {}'.format(proc.stderr))

    # get pdf path
    key_list = input_key.split('.')
    pdf_path = "/tmp/"+input_key.replace(key_list[-1], 'pdf')

    # put S3 Object
    if os.path.exists(pdf_path):
        print('PDF: {}'.format(pdf_path.replace("/tmp/", "")))
        print('Size: {}'.format(os.path.getsize(pdf_path)))
        data = open(pdf_path, 'rb')
        s3.put_object(Bucket=output_bucket, Key=pdf_path.replace("/tmp/", ""),Body=data)
        data.close()
    else :
        print("The PDF file({}) cannot be found".format(pdf_path))

    return ''
```


Lambda > Functions > document-conversion > Edit layers
Edit layers
Function runtime settings
Runtime
Python 3.8
Architecture
x86_64
Layers Info
Merge earlier
Merge later
Remove
Merge order
Name
Layer version
1
brotli
2
2
libreoffice-brotli
1
Cancel
Save


Amazon S3 > Buckets > document-storage-a1
document-storage-a1 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (2) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
college-details.xlsx
xlsx
May 25, 2024, 14:36:10 (UTC+05:30)
10.6 KB
Standard
DailyReport.pdf
pdf
May 25, 2024, 14:35:29 (UTC+05:30)
24.3 KB
Standard


https://ap-south-1.console.aws.amazon.com/s3/buckets/document-pdf-a2?region=ap-south-1&bucketType=general&tab=objects
Amazon S3 > Buckets > document-pdf-a2
document-pdf-a2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (3) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
ch4.pdf
pdf
May 25, 2024, 14:33:47 (UTC+05:30)
12.4 KB
Standard
college-details.pdf
pdf
May 25, 2024, 14:36:13 (UTC+05:30)
13.8 KB
Standard
DailyReport.pdf
pdf
May 25, 2024, 14:35:31 (UTC+05:30)
24.3 KB
Standard
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Code:-

```
import subprocess
import tarfile
import sys
from io import BytesIO
import os
import time
import boto3
import urllib.parse
import json

sys.path.append("/opt/brotli")
import brotli

ACCESS_KEY = "AKIAYBBJNKIEYBERKW4H"
SECRET_ACCESS_KEY = "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"

def extract_libre_office():
        buffer = BytesIO()
        with open('/opt/lo.tar.br', mode='rb') as fout:
            file = fout.read()
            buffer.write(brotli.decompress(file))
            buffer.seek(0)
            with tarfile.open(fileobj=buffer) as tar:
                tar.extractall('/tmp')

def lambda_handler(event, context):
    if os.path.exists("/tmp/instdir/program/soffice.bin"):
        pass
```

```
    else :
        # load libre
        extract_libre_office()


    # Get Trigger event
    if 'Records' in event.keys():
        input_bucket = event['Records'][0]['s3']['bucket']['name']
        output_bucket = "document-pdf-a2"
        input_key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
encoding='utf-8')
        s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY,
aws_secret_access_key=SECRET_ACCESS_KEY)
    else :
        return 'test finished'


    # get S3 Object
    file_path = '/tmp/'+input_key
    s3.download_file(input_bucket, input_key, file_path)


    # Document -> pdf
    proc = subprocess.run("/tmp/instdir/program/soffice.bin --headless --norestore --invisible -
-nodefault --nofirststartwizard --nolockcheck --nologo --convert-to pdf:writer_pdf_Export --
outdir /tmp {}".format("/tmp/"+input_key), shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
    print('STDOUT: {}'.format(proc.stdout))
    print('STDERR: {}'.format(proc.stderr))


    # get pdf path
    key_list = input_key.split('.')
    pdf_path = "/tmp/"+input_key.replace(key_list[-1], 'pdf')
```

```
    # put S3 Object
    if os.path.exists(pdf_path):
        print('PDF: {}'.format(pdf_path.replace("/tmp/", "")))
        print('Size: {}'.format(os.path.getsize(pdf_path)))
        data = open(pdf_path, 'rb')
        s3.put_object(Bucket=output_bucket, Key=pdf_path.replace("/tmp/", ""),Body=data)
        data.close()
    else :
        print("The PDF file({}) cannot be found".format(pdf_path))

    return "
```

libreoffice aws arn :- arn:aws:lambda:ap-south-1:764866452798:layer:libreoffice-brotli:1