

## Creating lambda function code locally and importing it

```
GNU nano 7.2 lambda_func
import os, io, json, boto3, logging
from PIL import Image

logger = logging.getLogger()
logger.setLevel(logging.INFO)
s3 = boto3.client('s3')

SIZES = {
    'desktop': (1920,1080),
    'tablet': (1280,800),
    'phone': (768,480),
}

def lambda_handler(event, context):
    logger.info("EVENT: " + json.dumps(event))
    rec = event['Records'][0]['s3']
    b = rec['bucket']['name']
    k = rec['object']['key']

    if not k.startswith('resized/'):
        logger.info(f"skip {k}")
        return

    obj = s3.get_object(Bucket=b, Key=k)
    img = Image.open(io.BytesIO(obj['Body'].read()))
    name, ext = os.path.splitext(os.path.basename(k))

    for folder, size in SIZES.items():
        im = img.copy()
        im.thumbnail(size, Image.LANCZOS)
        buf = io.BytesIO()
        im.save(buf, img.format)
        buf.seek(0)
        dst = f"{folder}/{name}{ext}"
        s3.put_object(Bucket=b, Key=dst, Body=buf, ContentType=obj['ContentType'])
        logger.info(f"wrote {dst}")

    return {'statusCode':200}
```

using predefined pillow layer image arn to read image

Layers <a href="#">Info</a>						<a href="#">Edit</a>	<a href="#">Add a layer</a>
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN		
1	Klayers-p310-Pillow	11	python3.10	x86_64	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p310-Pillow:11		

give lambda role necessary policies

## Modify permissions in s3Access-Logs [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements

### Policy editor

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "logs:CreateLogStream",
9         "logs:CreateLogGroup",
10        "logs:PutLogEvents"
11      ],
12      "Resource": "*"
13    },
14    {
15      "Sid": "VisualEditor1",
16      "Effect": "Allow",
17      "Action": "s3:GetObject",
18      "Resource": "arn:aws:s3:::shehab-s3-lambda/*"
19    },
20    {
21      "Sid": "VisualEditor2",
22      "Effect": "Allow",
23      "Action": "s3:PutObject",
24      "Resource": "arn:aws:s3:::shehab-s3-lambda/*"
25    }
26  ]
27 }
```

create an event notification to trigger lambda on uploading to s3

**Event notifications (1)**

EditDeleteCreate event notification

Send a notification when specific events occur in your bucket. [Learn more](#)

<input type="checkbox"/>	Name	Event types	Filters	Destination type	Destination
<input type="checkbox"/>	OnUploadTrigger	All object create events	resized/	Lambda function	<a href="#">shehab-image-lambda</a>

Results

on uploading an image to s3 in folder resized, 3 more folder with appropriate size were create

shehab-s3-lambda

Info

Objects

Metadata

Properties

Permissions

Metrics

Management





Access Poin

Objects (4)

Copy S3 URIDiv>

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	 <a href="#">desktop/</a>	Folder	-
<input type="checkbox"/>	 <a href="#">phone/</a>	Folder	-
<input type="checkbox"/>	 <a href="#">resized/</a>	Folder	-
<input type="checkbox"/>	 <a href="#">tablet/</a>	Folder	-

desktop/

Objects

Properties

Objects (1)

Objects are the fundamenta

Find objects by prefix

<input type="checkbox"/>	Name
<input type="checkbox"/>	 <a href="#">cat-s3.png</a>

