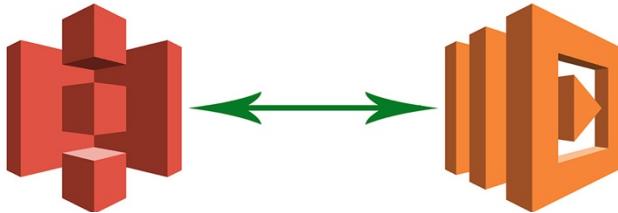


Final Project – CS 79E | Mehmet Karatas

Project Name: Lambda – S3 interaction

Summary: When uploading an image to S3 bucket, it triggers to lambda and lambda function start to resize the image as a thumbnail and save it to another S3 bucket.



Steps:

- Creating two S3 buckets.
- Creating Lambda Function.
- Configuring S3 bucket as a Lambda Event source.
- Triggering Lambda Function by uploading to S3.
- Monitor Lambda S3 functions through Amazon CloudWatch logs.

First, I started to create two S3 buckets with the name of **images-999888** and **images-999888-resized**. Then I uploaded the image **HappyFace.jpg** to the bucket named **images-999888**.

A screenshot of the AWS S3 console. The left sidebar shows navigation links for 'Amazon S3', 'Buckets' (which is selected), 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main content area is titled 'S3 buckets' and contains a search bar and filter options ('All access types'). It lists four buckets: 'images-999888' (Bucket and objects not public, US West (Oregon), Dec 18, 2019 3:28:18 PM GMT-0600), 'images-999888-resized' (Bucket and objects not public, US West (Oregon), Dec 18, 2019 3:28:36 PM GMT-0600), 'ql-cf-templates-1576704117-59b218849f252690-us-west-2' (Objects can be public, US West (Oregon), Dec 18, 2019 3:22:00 PM GMT-0600), and 'qltrail-lab-2284-1576704121' (Objects can be public, US East (N. Virginia), Dec 18, 2019 3:22:02 PM GMT-0600). At the bottom, there are tabs for 'Operations' (0 In progress, 2 Success, 0 Error) and links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

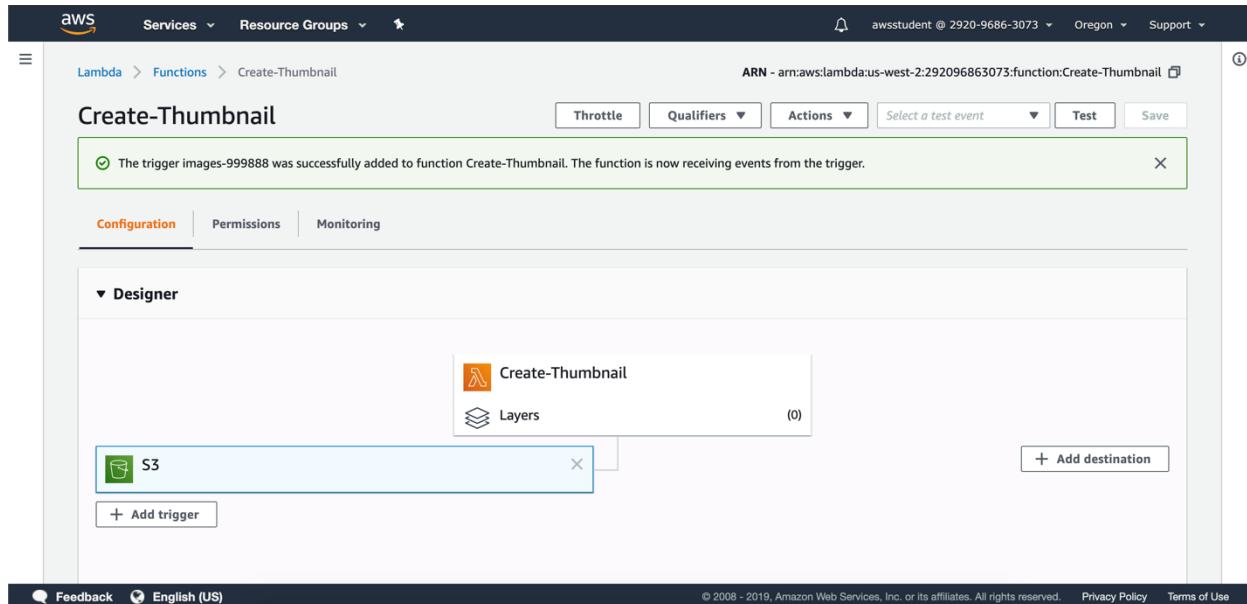
After creating two S3 buckets, I uploaded the big size image HappyFace.jpg to the bucket named images-999888.

The screenshot shows the AWS S3 console. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'awsstudent @ 2920-9686-3073'. Below the navigation bar, the path 'Amazon S3 > images-999888' is shown. The main content area displays a bucket named 'images-999888'. A search bar at the top says 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. To the right, it shows the region 'US West (Oregon)' with a location pin icon. A table lists one item: 'HappyFace.jpg'. The table columns are 'Name', 'Last modified', 'Size', and 'Storage class'. The file was last modified on Dec 18, 2019, at 3:29:42 PM GMT-0600, is 128.2 KB, and is in the Standard storage class. At the bottom of the page, there's a dark footer bar with 'Operations' and status indicators: '0 In progress', '2 Success', and '0 Error'.

Then I navigated to Lambda in us-west (Oregon) region. I clicked create function with the option Author from scratch. Function name: Create-Thumbnail and Runtime: Python 3.7 with lambda-execution-role.

The screenshot shows the AWS Lambda console. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'awsstudent @ 2920-9686-3073'. Below the navigation bar, the path 'Lambda > Functions > Create-Thumbnail' is shown. The main content area displays a function named 'Create-Thumbnail'. A toolbar above the function includes 'Throttle', 'Qualifiers', 'Actions', 'Select a test event', 'Test', and 'Save'. Below the toolbar, there are tabs for 'Configuration', 'Permissions', and 'Monitoring'. The 'Configuration' tab is selected. Under the 'Configuration' tab, the 'Designer' section is open, showing a function block labeled 'Create-Thumbnail'. Below the function block, there's a 'Layers' section with '(0)' and a button '+ Add destination'. At the bottom left, there are buttons '+ Add trigger' and '+ Add destination'. At the bottom right, there are buttons '+ Add destination' and '+ Add destination'. The footer of the page includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

After this step, I clicked “Add trigger” and selected trigger as S3 with my bucket named **images-999888** and the Event type: All objects create events.



After this step I clicked, Create-Thumbnail.

Code entry type: Upload file from Amazon S3

Runtime: Python 3.7

Handler: CreateThumbnail.handler

And I used this: <https://s3-us-west-2.amazonaws.com/us-west-2-aws-training/awsu-spl/spl-88/2.3.2.prod/scripts/CreateThumbnail.zip> provided Amazon S3 link URL by the project.

The code that was provided was like this:

```
import boto3
import os
import sys
import uuid
from PIL import Image
import PIL.Image

s3_client = boto3.client('s3')

def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
        image.thumbnail((128, 128))
        image.save(resized_path)

def handler(event, context):
    for record in event['Records']:
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']
        download_path = '/tmp/{}{}'.format(uuid.uuid4(), key)
        upload_path = '/tmp/resized-{}'.format(key)

        s3_client.download_file(bucket, key, download_path)
        resize_image(download_path, upload_path)
        s3_client.upload_file(upload_path, '{}-resized'.format(bucket), key)
```

After this step, it is time to test the function.

Event template: Amazon S3 Put

Event name: Upload

When I did that, a sample template displayed that shows the event data send to a Lambda function. I replaced the example bucket names with my bucket name images-999888 in the template and changed the test/key with my image name. Then I clicked create to create the test. AWS Lambda triggered my function using the images in my S3 bucket as an input.

When I clicked Test, I saw the success result.

The screenshot shows the AWS Lambda Test Results page for a function named 'Create-Thumbnail'. At the top, there is a success message: 'Execution result: succeeded (logs)'. Below this, a 'Details' section is expanded, showing a text area with the word 'null'. Under the 'Summary' section, it lists SHA-256 code hash, Request ID, Duration (1041.70 ms), Billed duration (1100 ms), and resources configured (128 MB). Max memory used is listed as 87 MB Init Duration: 446.12 ms. The 'Log output' section contains CloudWatch log entries:

```
START RequestId: ceb70ff9-a159-4d99-b006-7a79a158fa1b Version: $LATEST
END RequestId: ceb70ff9-a159-4d99-b006-7a79a158fa1b
REPORT RequestId: ceb70ff9-a159-4d99-b006-7a79a158fa1b Duration: 1041.70 ms Billed Duration: 1100 ms Memory Size: 128 MB
Max Memory Used: 87 MB Init Duration: 446.12 ms
```

At the bottom of the page, there are links for Feedback, English (US), Copyright notice (© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

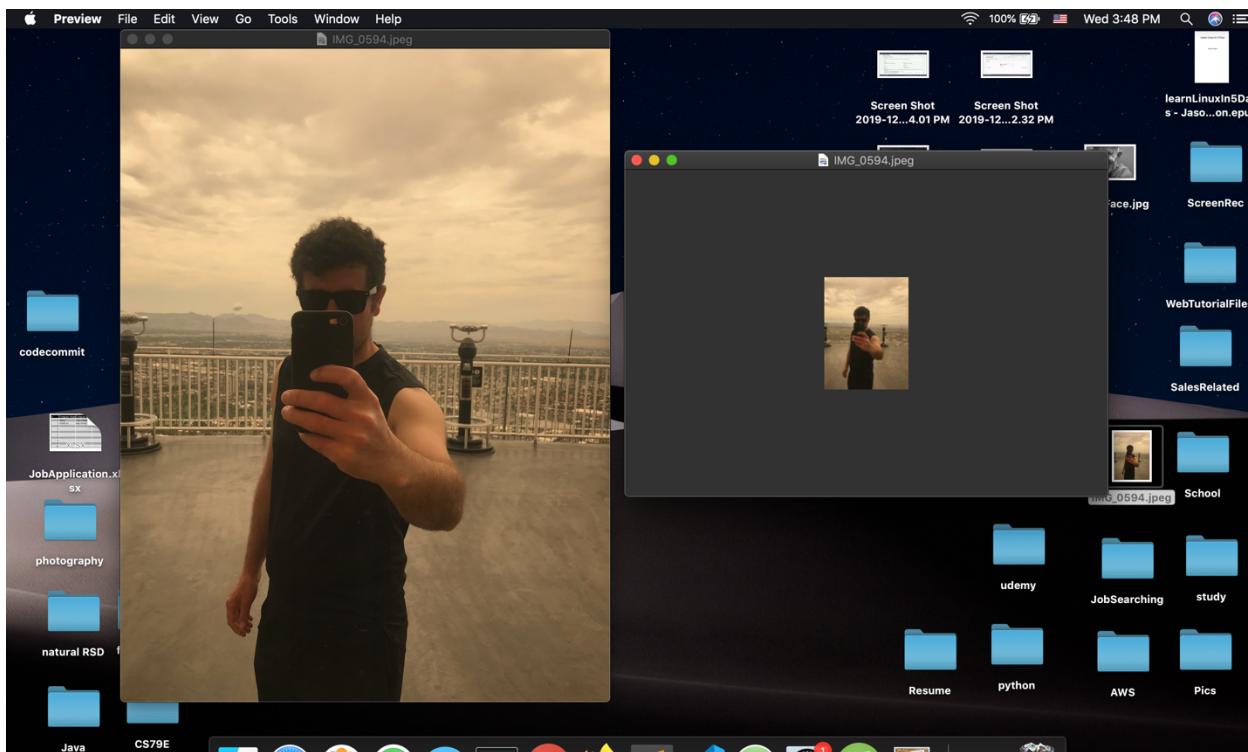
When I clicked to extend the details from the screenshot, we can see duration, memory usage, request ID, log output and etc. After this step I navigated to S3 to check the other bucket that I created earlier with the name of **images-999888-resized**. And I saw that the picture had been saved there as a thumbnail as seen in the screen shot. As it can be seen, the size is 2.6 KB only and the dimensions are 128 X 85. It is obviously much smaller size. The image is now the smaller thumbnail of the original image.

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), a bell icon, 'awsstudent @ 2920-9686-3073' (dropdown), 'Global' (dropdown), and 'Support' (dropdown). Below the navigation bar, the path 'Amazon S3 > images-999888-resized > HappyFace.jpg' is displayed. The main content area shows the file 'HappyFace.jpg' with the status 'Latest version'. Below the file name are tabs for 'Overview' (selected), 'Properties', 'Permissions', and 'Select from'. Underneath these tabs are buttons for 'Open', 'Download', 'Download as', 'Make public', and 'Copy path'. The detailed properties section includes:

- Owner:** aws032544
- Last modified:** Dec 18, 2019 3:43:50 PM GMT-0600
- Etag:** 06669e6fbde55b5a8e77f61fef9a2661
- Storage class:** Standard
- Server-side encryption:** None
- Size:** 2.6 KB
- Key:** HappyFace.jpg
- Object URL:** <https://images-999888-resized.s3-us-west-2.amazonaws.com/HappyFace.jpg>

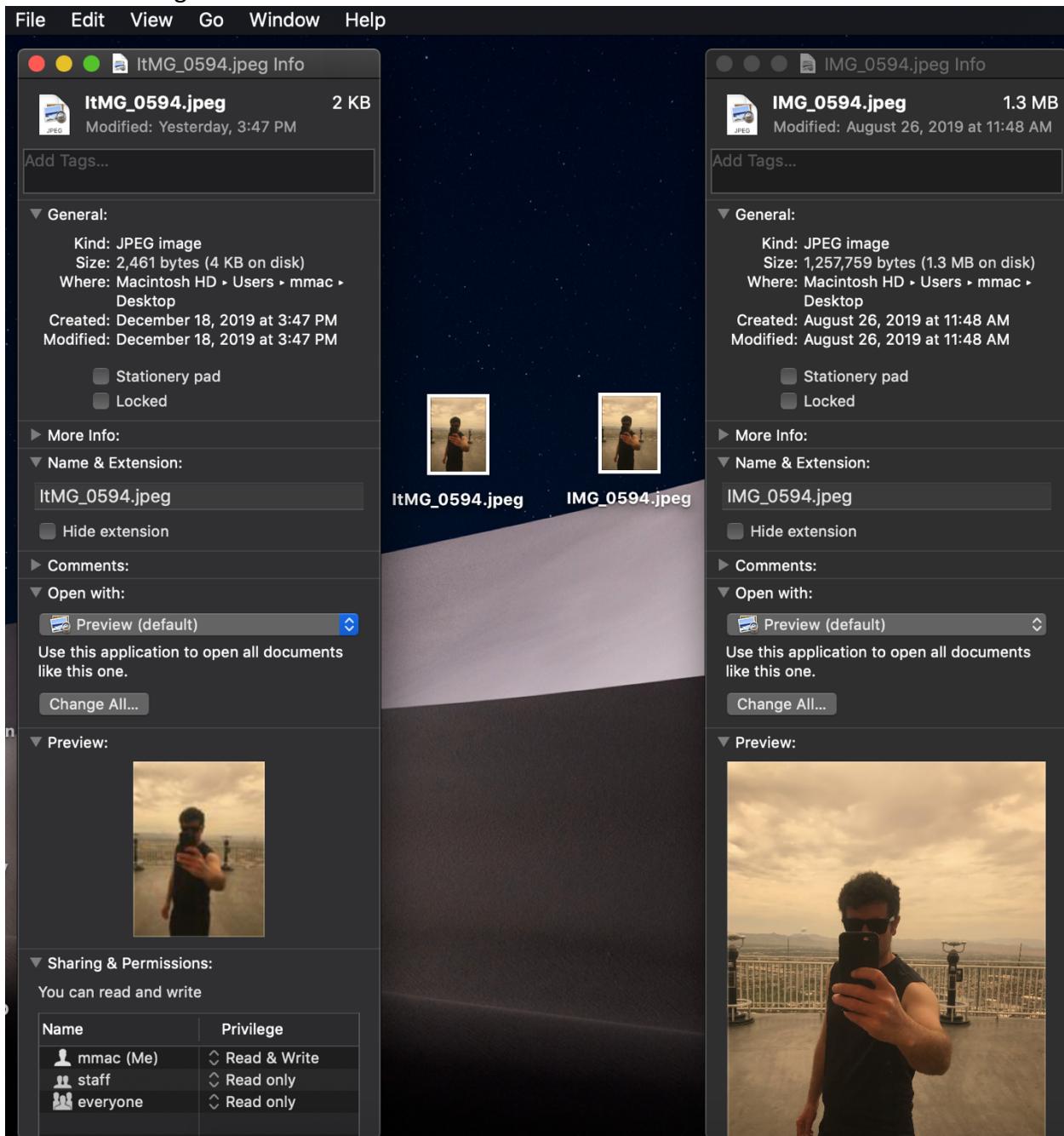
At the bottom of the properties section, there are buttons for 'Operations' (0 In progress, 2 Success, 0 Error) and links for 'Feedback' and 'English (US)'. The footer contains the copyright notice '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links for 'Privacy Policy' and 'Terms of Use'.

After I saw that everything went through with success, it was the time for uploading my own pictures to my main S3 bucket with **images-99988** name.

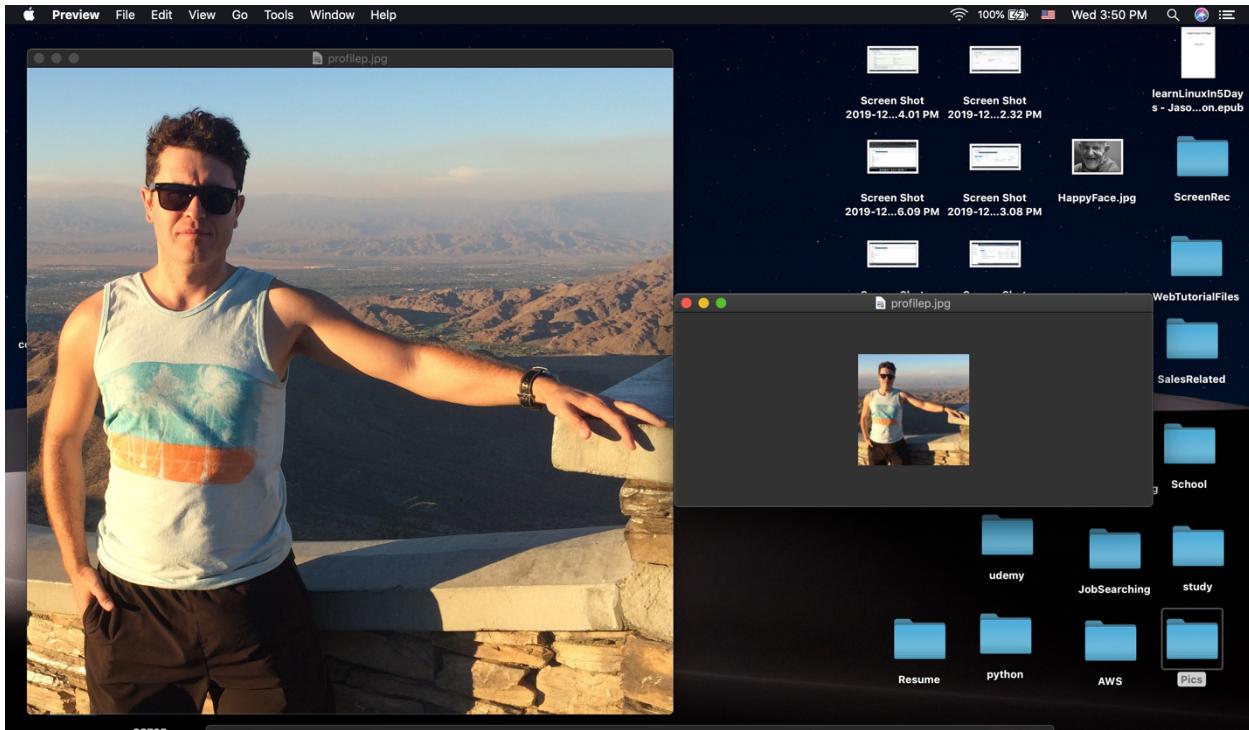


After I uploaded my picture to images-999888 bucket I checked images-999888-resized bucket and I saw my own picture saved there as a thumbnail also. Then I downloaded my resized image and open both side by side. That means that my project was functioning with a great success.

Then I checked the info for both images to see the difference in size and quality also as can be seen at the image below.



Whenever I was uploading any images to my S3 bucket, it was triggering the lambda and the lambda was executing the lambda function to resize my image and save it to another bucket that I created for resized images. I tried another picture of me and it also went through without any issue.



As it can be seen the same thing was happening. The lambda function was keep working whenever I was uploading any images to my images-999888 bucket.

Then I uploaded multiple images at once like it can be seen at screenshot below:

Name	Last modified	Size	Storage class
HappyFace.jpg	Dec 18, 2019 3:29:42 PM GMT-0600	128.2 KB	Standard
IMG_0594.jpeg	Dec 18, 2019 3:46:54 PM GMT-0600	1.2 MB	Standard
IMG_20171030_135716.jpg	Dec 18, 2019 3:52:18 PM GMT-0600	1.9 MB	Standard
IMG_20171030_135732.jpg	Dec 18, 2019 3:52:19 PM GMT-0600	3.0 MB	Standard
IMG_20171030_135747.jpg	Dec 18, 2019 3:52:16 PM GMT-0600	2.0 MB	Standard
IMG_20171030_140306.jpg	Dec 18, 2019 3:52:17 PM GMT-0600	4.2 MB	Standard
IMG_20171030_140313.jpg	Dec 18, 2019 3:52:16 PM GMT-0600	3.9 MB	Standard
IMG_20171030_140314.jpg	Dec 18, 2019 3:52:15 PM GMT-0600	3.9 MB	Standard
IMG_20171030_140330.jpg	Dec 18, 2019 3:52:09 PM GMT-0600	3.5 MB	Standard
IMG_20171030_140405.jpg	Dec 18, 2019 3:52:10 PM GMT-0600	4.3 MB	Standard
IMG_20171030_151506.jpg	Dec 18, 2019 3:52:06 PM GMT-0600	1.7 MB	Standard
IMG_20171030_151513.jpg	Dec 18, 2019 3:52:10 PM GMT-0600	1.9 MB	Standard
IMG_20171030_151521.jpg	Dec 18, 2019 3:52:06 PM GMT-0600	1.6 MB	Standard

Operations 0 In progress 1 Success 0 Error

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

When we check the sizes, we can clearly see that they are in MBs. After the upload was successfully done I went to check another bucket named **images-999888-resized** and I saw that all these images are saved there as thumbnails also as can be seen below.

The screenshot shows the AWS S3 console interface. At the top, there's a search bar with placeholder text "Type a prefix and press Enter to search. Press ESC to clear." Below the search bar are buttons for "Upload", "+ Create folder", "Download", and "Actions". The location "US West (Oregon)" is shown. A table lists 16 files, all of which are Standard type and have sizes between 2.4 KB and 4.1 KB. The files were uploaded on December 18, 2019, at various times between 3:43:50 PM and 3:49:21 PM. The last file listed is "profilepic.jpg".

We can clearly see that the image sizes are smaller. Also, the lambda function was working without any issue with multiple uploads.

After all of these tests I checked the monitoring for lambda function as can be seen below.

The screenshot shows the AWS Lambda function monitoring page for the "Create-Thumbnail" function. The top navigation bar includes "Services", "Resource Groups", "Feedback", "English (US)", and "awsstudent @ 2920-9686-3073". The main section is titled "Create-Thumbnail" and has tabs for "Configuration", "Permissions", and "Monitoring". The "Monitoring" tab is selected. It displays three line charts: "Invocations, ProvisionedConcurrencyInvoca...", "Duration", and "Error count and success rate (%)".

- Invocations, ProvisionedConcurrencyInvoca...**: Shows two series: "Invocations" (blue line) and "ProvisionedConcurrency/Invocations" (orange line). The blue line shows a sharp increase from 1 to 11 at 21:45.
- Duration**: Shows three series: "Duration Minimum" (blue), "Duration Average" (orange), and "Duration Maximum" (green). All three series show a sharp increase starting around 21:40.
- Error count and success rate (%)**: Shows two series: "Errors" (red line) and "Success rate (%)" (green line). The red line is at 0, and the green line is at 100%.

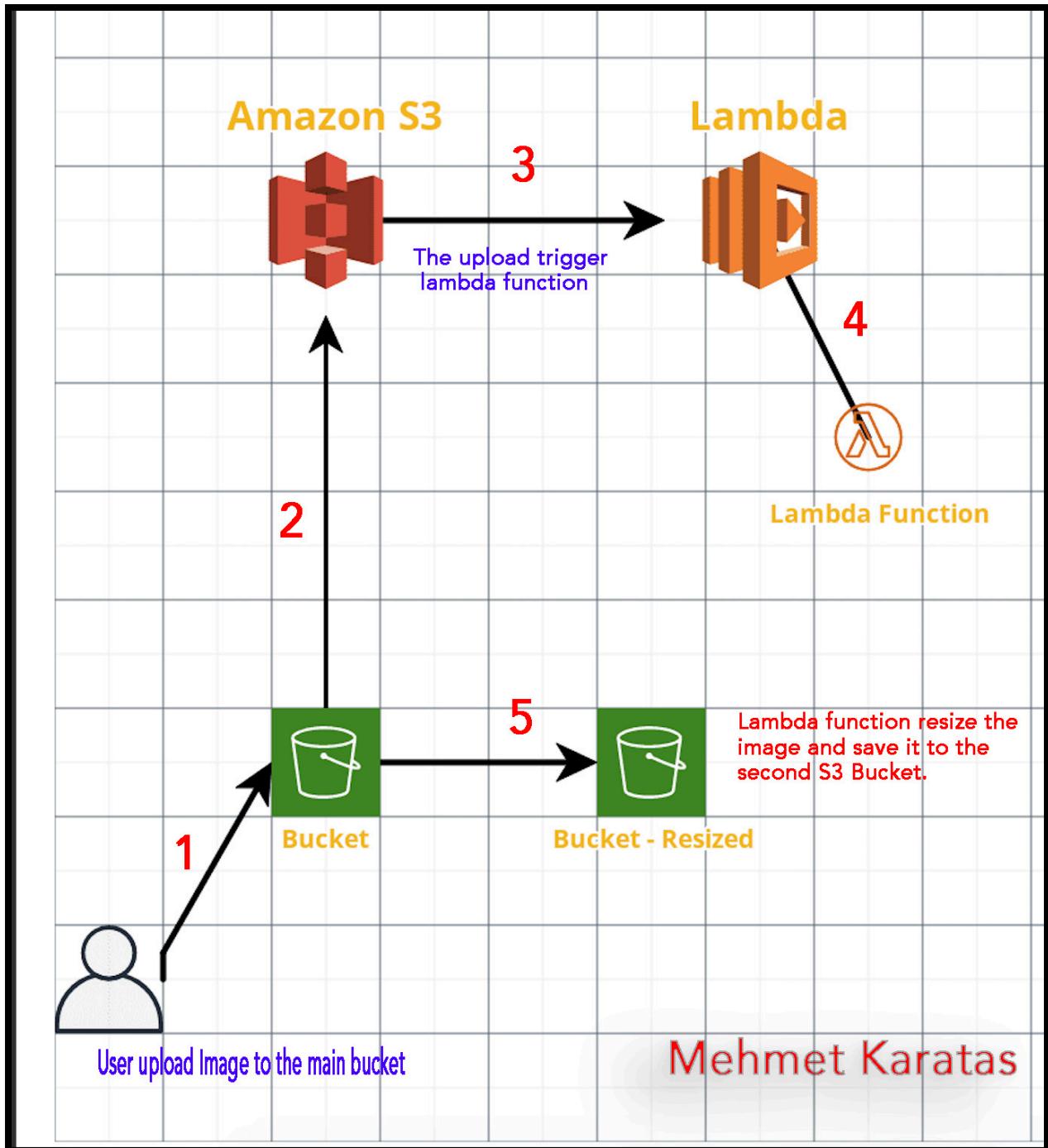
Below the charts, there are sections for "Throttles" and "IteratorAge". The bottom of the page includes "Feedback", "English (US)", and standard AWS footer links.

I checked it in CloudWatch too with clicking view logs in CloudWatch. This below screenshot shows the CloudWatch logs.

The screenshot shows the AWS CloudWatch Logs interface. The left sidebar is titled "Logs" and includes options like Log groups, Insights, Metrics, Events, Rules, Event Buses, ServiceLens, Traces, Contributor Insights, Settings, and Favorites. The main pane displays log entries for a Lambda function named "Create-Thumbnail". The log entries are timestamped at 2019-12-18 21:52:10. The first entry is a START request with RequestId 4a249794-f66a-4efd-b58a-325fab43a49c and Version \$LATEST. Subsequent entries show END requests, REPORT requests, and another START request. The logs provide detailed metrics such as Duration, Billed Duration, Memory Size, and Max Memory Used.

Time (UTC +00:00)	Message
2019-12-18 21:52:10	START RequestId: 4a249794-f66a-4efd-b58a-325fab43a49c Version: \$LATEST
21:52:13	END RequestId: 4a249794-f66a-4efd-b58a-325fab43a49c
21:52:13	REPORT RequestId: 4a249794-f66a-4efd-b58a-325fab43a49c Duration: 2756.13 ms Billed Duration: 2800 ms Memory Size: 128 MB Max Memory Used: 92 MB Init Duration: 441.21 ms
21:52:14	START RequestId: 4768d606-b926-46f0-b691-1cfbc4144157 Version: \$LATEST
21:52:16	END RequestId: 4768d606-b926-46f0-b691-1cfbc4144157
21:52:16	REPORT RequestId: 4768d606-b926-46f0-b691-1cfbc4144157 Duration: 1861.94 ms Billed Duration: 1900 ms Memory Size: 128 MB Max Memory Used: 97 MB
	REPORT RequestId: 4768d606-b926-46f0-b691-1cfbc4144157 Duration: 1861.94 ms Billed Duration: 1900 ms Memory Size: 128 MB Max Memory Used: 97 MB

I tried to show it as a diagram using the tool provided by [cloudcraft.co](#) and photoshop. My diagram can be seen below.



With this project, I did the followings:

- Created an AWS Lambda function
- Configured S3 bucket as a Lambda Event Source
- Triggered function by uploading an object to Amazon S3
- Monitored AWS Lambda S3 functions through Amazon CloudWatch logs.