AWS Lambda and S3 implementation in django_checklist project:

S3 management console: https://console.aws.amazon.com/s3/buckets/django-checklist-files/profile_pics/?region=us-east-1&tab=overview

IAM page: https://console.aws.amazon.com/iam/home#/home

Lambda functions page: https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Tutorial page: https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html

I wanted to create a lambda function to reduce the size of the image as it is uploaded so that I don't fill my S3 bucket storage quickly and go over the AWS free tier limits.

Step 0: Configure amazon AWS CLI to run the commands in the following steps. The links and steps originate from the link mentioned in the pre-requisites section of the tutorial link. Refer this link to install and verify AWS CLI. https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-mac.html#cliv2-mac-install-cmd-all-users. Now, configure AWS credentials. I created another user for this but that didn't work so I provided the credentials of the first user created.

Step 1: I referred this link: https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html to create a lambda function. In creating a IAM role, I also provided AWSLambdaFullAccess and AWSLambdaBasicExecutionRole in addition to the AWSLambdaExecute policy mentioned in the tutorial.

Step 2: To create a Lambda function, I followed this link: https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example-deployment-pkg.html#with-s3-example-deployment-pkg-python. The code there reduces the dimensions of an image by half when it is uploaded. The modifications made to the code were such:

A) resize_image() function was modified to simply reduce the dimensions of the image only if any of the height or width of the image are more than 300 pixels.

B) s3_client.upload_file() was modified to upload the file to the same bucket with the same name.

C) When we will deploy this lambda function, we will configure it to execute whenever an object is placed in the S3 bucket. Now, if we don't have a boolean condition, we won't be able to stop infinite execution of our function. Hence, we return a boolean value and only upload the new reduced file if the image was actually resized and the boolean value returned by resize_image() was True.

Step 3: Now, we need to create a deployment package. I followed the steps given in the link of step 2 that showed the sample lambda function creation. However, since MacOS python distribution is different than that of an AWS machine OS which is Amazon Linux, I ran into errors while invoking function. Hence, I opened up an EC2 machine (under free tier) and followed the steps on this OS. I created the function.zip and then SCP'd it to my local machine. Once, my function.zip is ready, I follow the same steps in the link of step 1 to create the function and invoke it.

The commands execution is as follows: [https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html]

1. aws lambda delete-function --function-name CreateThumbnail **[if a function is already created]**

2. zip -g function.zip lambda_function.py

3. aws lambda create-function --function-name CreateThumbnail --zip-file fileb://function.zip --handler lambda_function.lambda_handler --runtime python3.7 --timeout 10 --memory-size 1024 --role arn:aws:iam::<USER_ARN>:role/lambda-role

4. aws lambda invoke --function-name CreateThumbnail --invocation-type RequestResponse --payload file://inputFile.txt --cli-binary-format raw-in-base64-out outputfile.txt

OR

4. aws lambda invoke --function-name CreateThumbnail --invocation-type RequestResponse --payload file://inputFile.txt outputfile.txt

The create function had a minor change in the handler name. The handler name convention is as follows: part before "." Is filename (lambda_function.py in our case) and part after "." is handler function name (lambda_handler in our case). The runtime was changed to python3.7. The role was modified by copying the **lambda-role** role's ARN created in Step 1.

The invoke function also resulted in some error for binary data conversion. I searched for the error and based on this link: https://forums.aws.amazon.com/thread.jspa?threadID=317581 added the part **"--cli-binary-format raw-in-base64-out"** to invoke command.

Step 4: Configure S3 to publish events so our lambda function is executed on an event. Follow the steps in the link in step 1 (tutorial link). Complete the steps following it completely. You should have a working implementation of the lambda function that is executed on object creation event in S3 bucket.

5. aws lambda add-permission --function-name CreateThumbnail --principal s3.amazonaws.com --statement-id s3invoke --action "lambda:InvokeFunction" --source-arn arn:aws:s3:::django-checklist-files --source-account <USER_ARN>

This experience was worth the effort of 2 days, every bit satisfying when it works.