LocalStack Tutorial

Introduction

This is an app to resize images uploaded to S3 in a serverless way. A simple web fronted using HTML and JavaScript provides a way for users to upload images that are resized and listed. We use a Lambda to generate S3 pre-signed URLs so the upload form can upload directly to S3 rather than going through the Lambda. S3 bucket notifications are used to trigger a Python Lambda that runs image resizing. Another Lambda is used to list all uploaded and resized images, and provide pre-signed URLs for the browser to display them.

Here's a short summary of AWS service features we use:

- S3 bucket notifications to trigger a Lambda
- S3 pre-signed POST
- S3 website
- Lambda function URLs

> The tutorial original source is: https://github.com/localstack-samples/sample-serverless-image-resizer-s3-lambda

Running the serverless image resizer app

> Please note all the commands assume you are located in the `tutorial` directory, you already setup the environment following the provided guide (Python 3.11 and virtualenv, Docker, LocalStack and utilities), and you activated your Python virtualenv.

Create the buckets

The names are completely configurable via SSM:

```
awslocal s3 mb s3://localstack-thumbnails-app-images
awslocal s3 mb s3://localstack-thumbnails-app-resized
```

Put the bucket names into the parameter store

```
awslocal ssm put-parameter --name /localstack-thumbnail-app/buckets/images
--type "String" --value "localstack-thumbnails-app-images"
awslocal ssm put-parameter --name /localstack-thumbnail-app/buckets/resized
--type "String" --value "localstack-thumbnails-app-resized"
```

Create the lambdas

S3 pre-signed POST URL generator

This Lambda is responsible for generating pre-signed POST URLs to upload files to an S3 bucket.

> Please note, change `zip` with `tar` if needed on Windows!

```
(cd lambdas/presign; rm -f lambda.zip; zip lambda.zip handler.py)
awslocal lambda create-function \
    --function-name presign \
    --runtime python3.11 \
    --timeout 10 \
    --zip-file fileb://lambdas/presign/lambda.zip \
    --handler handler.handler \
    --role arn:aws:iam::000000000000:role/lambda-role \
    --environment Variables="{STAGE=local}"
```

Create the function URL:

```
awslocal lambda create-function-url-config \
    --function-name presign \
    --auth-type NONE
```

Image lister lambda

The `list` Lambda is very similar:

```
(cd lambdas/list; rm -f lambda.zip; zip lambda.zip handler.py)
awslocal lambda create-function \
    --function-name list \
    --handler handler.handler \
    --zip-file fileb://lambdas/list/lambda.zip \
    --runtime python3.11 \
    --role arn:aws:iam::000000000000:role/lambda-role \
    --environment Variables="{STAGE=local}"
```

Create the function URL:

```
awslocal lambda create-function-url-config \
    --function-name list \
    --auth-type NONE
```

Resizer Lambda

```
(
    cd lambdas/resize
```

```
    rm -rf package lambda.zip
    mkdir package
    pip install -r requirements.txt -t package --platform
manylinux2014_x86_64 --only-binary=:all:
    zip lambda.zip handler.py
    cd package
    zip -r ../lambda.zip *;
)
awslocal lambda create-function \
    --function-name resize \
    --runtime python3.11 \
    --timeout 10 \
    --zip-file fileb://lambdas/resize/lambda.zip \
    --handler handler.handler \
    --role arn:aws:iam::000000000000:role/lambda-role \
    --environment Variables="{STAGE=local}"
```

Connect the S3 bucket to the resizer lambda

```
awslocal s3api put-bucket-notification-configuration \
    --bucket localstack-thumbnails-app-images \
    --notification-configuration "{\"LambdaFunctionConfigurations\":
[{\"LambdaFunctionArn\": \"$(awslocal lambda get-function --function-name
resize | jq -r .Configuration.FunctionArn)\", \"Events\":
[\"s3:ObjectCreated:*\"]}]}"
```

Create the static s3 webapp

```
awslocal s3 mb s3://webapp
awslocal s3 sync --delete ./website s3://webapp
awslocal s3 website s3://webapp --index-document index.html
```

Using the application

Once deployed, visit http://webapp.s3-website.localhost.localstack.cloud:4566

Paste the Function URL of the presign Lambda you created earlier into the form field using the following commands (or click on the "**Load from API**" button).

```
awslocal lambda list-function-url-configs --function-name presign
awslocal lambda list-function-url-configs --function-name list
```

After uploading a file, you can download the resized file from the `localstack-thumbnails-app-resized` bucket.

Run integration tests

Once all resource are created on LocalStack, you can run the automated integration tests.

```
pytest tests/
```