

When learning about Lambda, you will often hear about “triggers”. This is because most use cases for serverless computing typically involve code that runs in response to an “event”. Imagine an application where users can upload an image of an object and get back links to buy that object online. The code that recognizes the object and queries e-commerce stores could very well be a Lambda function that gets “triggered” by the image upload.

Create a role

1. Sign in to your AWS console.
2. On the search bar at the top, search for “IAM” and click on it.
3. On the left side menu, click on “Roles”.
4. Click on “Create role”.
5. Select “AWS service” as the type of trusted entity.
6. Select “Lambda” as the use case.
7. Click on “Next: permissions”.
8. Tick the box of the policy “AdministratorAccess”.
9. Click on “Next: Tags”.
10. Click on “Next: Review”.
11. Set “LambdaAdminAccess” as the role name.
12. Click on “Create role”.

Create a Lambda function

1. Sign in to your AWS console.
2. Go to Services > Lambda.
3. Click on “Create function”.
4. Select “Author from scratch”.
5. Give your function a name you can recognize. Since we are just testing, we recommend something like “wbs-test”.
6. On Runtime, select “Python 3.8”.
7. On “Permissions”, click on “Change default execution role”, tick “Use an existing role” and select the “LambdaAdminAccess” role we just created.
8. Click on “Create function”.

Upload a Layer with Python modules to your Lambda function

AWS Data Wrangler + KLayers:

[AWS Data Wrangler](#) is a collection of modules relevant to data storage and manipulation (e.g. numpy, pandas...) that has been prepared to work seamlessly with AWS services. You could install it on your computer like a regular module (well, like a super-module), but what we want to do here is to make it available to your Lambda function. To do so, follow these instructions:

1. Go back to your Lambda function on the AWS console.
2. Scroll down to the bottom of the page. On the “Layers” section, click on “Add layer”
3. As you can see, there are 3 ways to add a layer to the function. Click on AWS Layers.
4. Select AWSDataWrangler-Python38.
5. Click on “Add”.

Now your Lambda function will have access to packages like Pandas. But one module is still likely missing to perform all of the actions you need your code to do: SQLAlchemy. Unfortunately, it is not present on AWS Data Wrangler. The most straight-forward solution here is to use the publicly available work of other great developers, like [Keith Rozario](#), who builds and maintains a list of Lambda Layers with some of the most popular Python modules. Go through the following simple steps:

1. Visit the GitHub repository from Keith about Klayers [here](#).
2. Find the list of all layer version arns available by region. You will find a list of csv files with AWS region names:
3. Click on the region where your Lambda function is located. If you don't remember it, just look go to your Lambda Function on the AWS Console and look at the top right. Next to your username, you'll see a region name (e.g. "Ohio", "Frankfurt"...). These regions have official names (e.g. us-east-2, eu-central-1...). Check the official name of your region [here](#).
4. Find SQLAlchemy on the long list of Python packages. Make sure to locate the "latest" version of the package and not a "deprecated" one.
5. Copy the arn code (the string that begins like `arn:aws:lambda:us-east-2:77321...`).
6. Go back to your Lambda Function on the AWS Console and add another Layer to your Lambda function.
7. This time, rather than selecting a layer from the list of "AWS Layers, you will have to click on "Specify an ARN" and paste the arn code you copied from the KLayers repository.
8. Click on "Add".