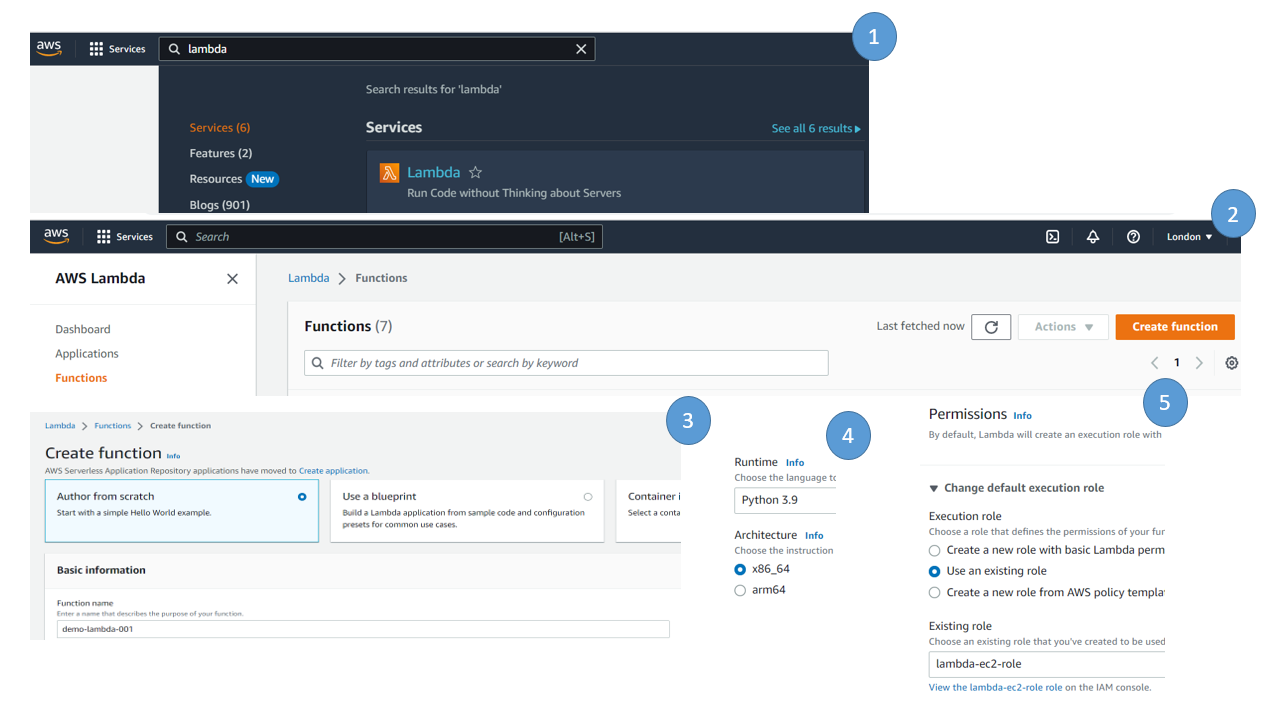
**Data Engineering Environment Set up — How to Run Lambda function Locally**

In Data engineering eco system, serverless services in AWS played a vital role. When we all are talking about AWS serverless services, the first service come to mind is AWS Lambda. AWS Lambda is the most popular serverless service provided by AWS. It is widely used in all most all application integrations whether api integration or event driven architecture or loading data (mini batch processing) cases etc.

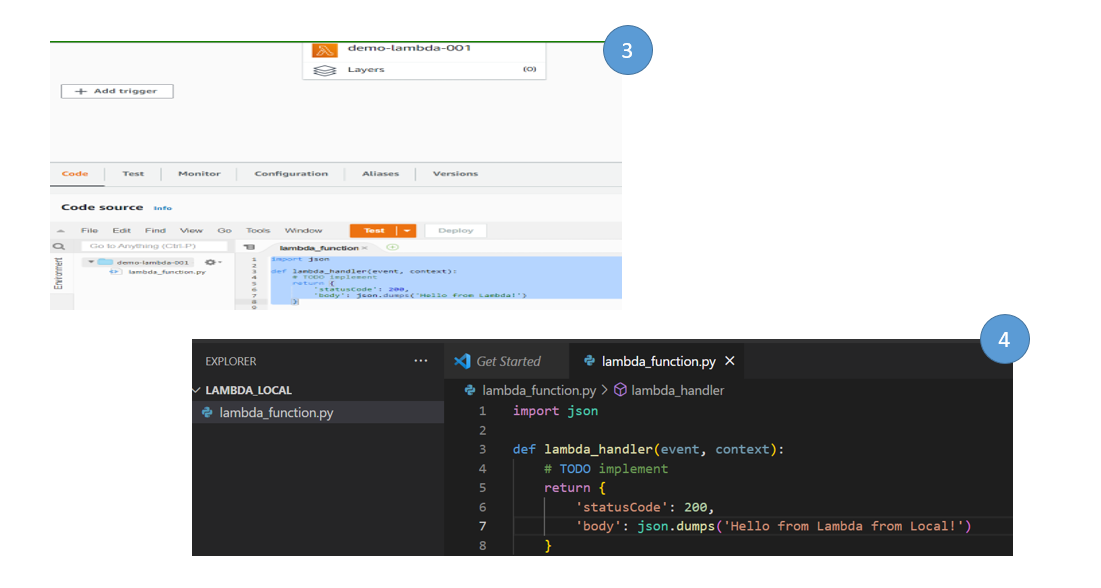
For a developer to develop lambda function directly in AWS console is not a good idea as he/she will perform changes to the development code very frequently and test it before final version is ready for end to end testing in AWS. In this blog, we will explain very easy step how we can create a lambda function locally, test it and then upload it in AWS. This is a simple example and this will be used in our future data engineering in AWS blogs.

Step -1: First create a lambda function in AWS (or copy the sample code). To do the same.

1. Login to AWS console and search for lambda
2. Click on Create function and give the function name as demo-lambda-001 (User can give any name)
3. Select the run time as python3.9 (Lambda supports many languages; the complete details are available in drop down)
4. The architecture is x86\_64 which is standard.
5. If user have any requirement to interact with any AWS services, create an appropriate role and assign require permission and attach the role to lambda function. In this case we created an IAM role which have full permission to ec2 and S3 and lambda execution role. The click create function to create the function.



Step -2: Open your local python environment, in this case we are VSCODE extension for developing the lambda function locally. For simple approach, we created a folder called lambda\_local and put all our python scripts in that folder. First create file called lambda\_function and copy the lambda function content from AWS lambda code ( which is generated by default).

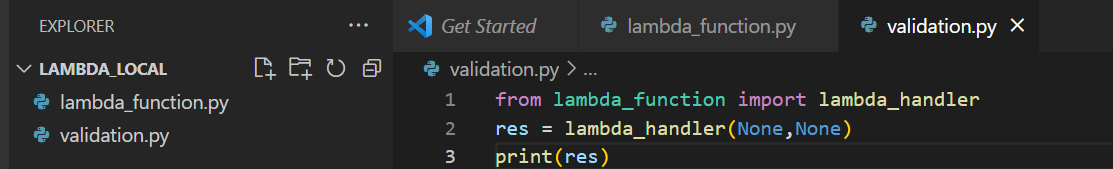


Note – We added couple of more words in the body part.

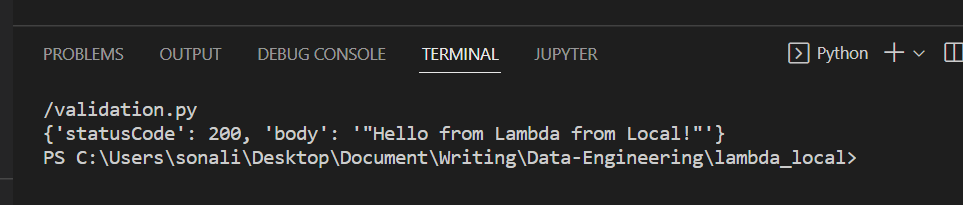
This way we completed the lambda handler creation locally.

Step 3: For testing lambda function locally, we will create a function called validation.py and called the lambda handler. To create the validation function, click new python file in the same folder.

The lambda handler requires 2 arguments event and context, for simplicity we make both arguments as None and capture the response. Finally, we print the response to ensure whether the function is working or not locally.



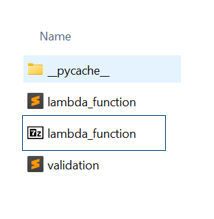
Now run the validation.py script to see whether we are able to run the lambda function or not. It will print the response message



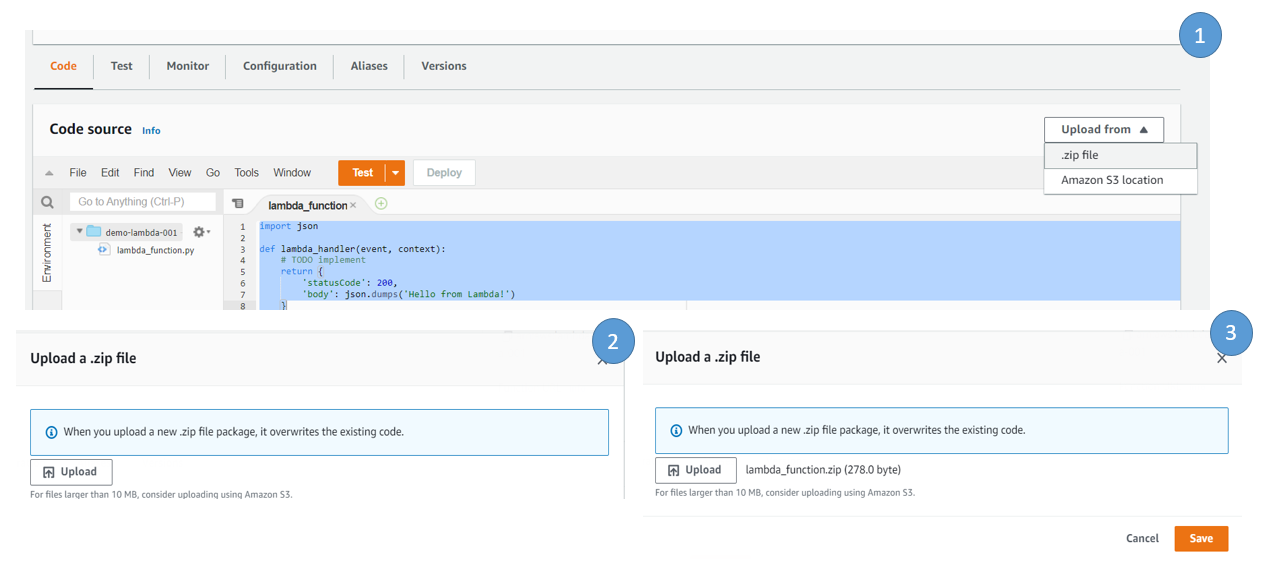
Congratulations, you are able to run a lambda function locally. Now you can put all complex logic in the lambda and test locally.

Step -4: To deploy this function in AWS,

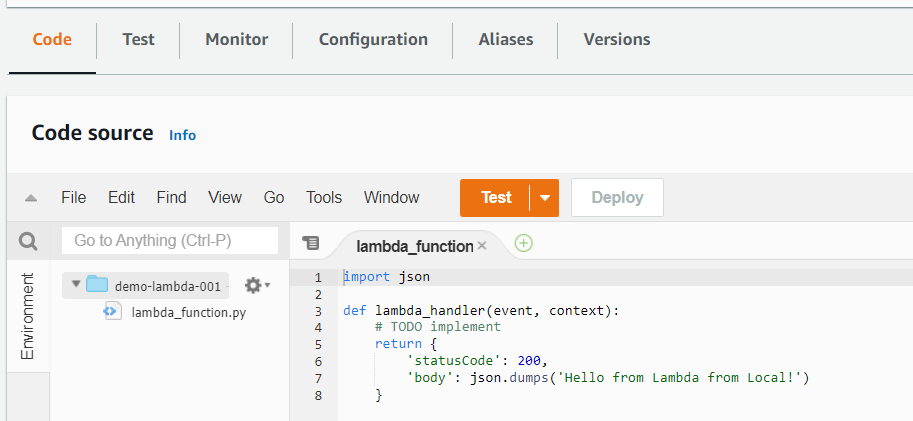
1. Zip the lambda function, to do the zip. Compress the lambda function. Since we are using VSCODE in windows, we compressed it manually



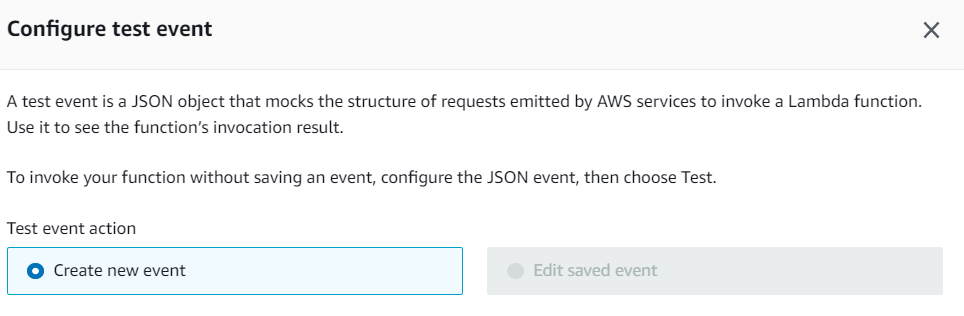
1. Go to AWS lambda console, select upload from, select .zip file
2. Click upload and select your lambda zip file and click on Save.



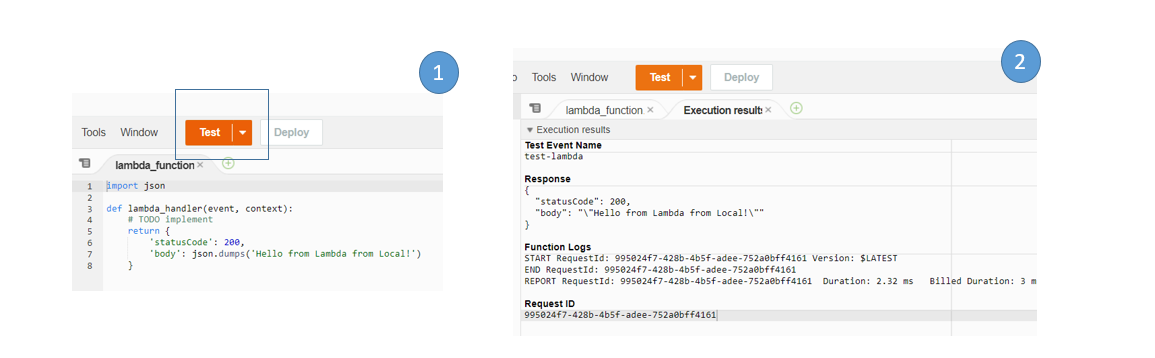
User can see the changes in the lambda code



Step -5: To Test the code from AWS console, create a test event, to do the same click on Test, and create a test event. Give any name, here we gave the name as test-lambda and save it



Step -6: To run the function, click on test again and it will execute the output result will be shown in execution\_results tab. This way user can develop lambda function, develop all complex logic and test it locally and then upload or deploy the code in AWS



Note – If user has the requirement to interact lots of aws services via lambda, we need to set up a profile and assign right permissions to the user profile and then using boto3, we can interact with different aws services. We are going to have data engineering blogs for this kind of activity in upcoming days.