**Aws Lambda Function packaging with serverless framework**

If you have Node and NPM installed, install the Serverless Framework globally with:

$ npm install -g serverless

You'll also need to configure your environment with AWS credentials.

We can create a service from a template. I'm going to use Python 3

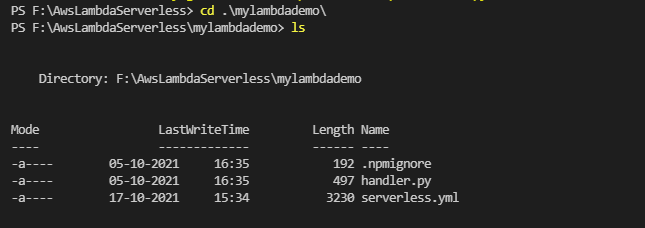
$ serverless create --template aws-python3 --name aws-lambda-stepfn --path mylambdademo



This will create a Serverless Python 3 template project at the given path (mylambdademo) .

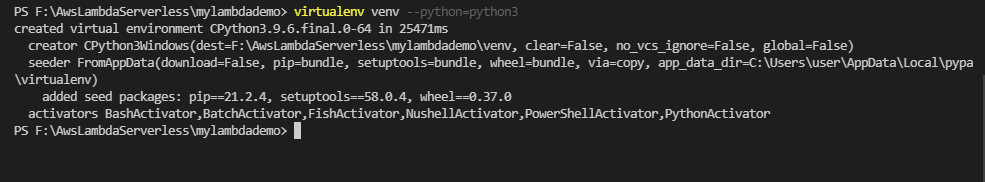
Go inside to the project

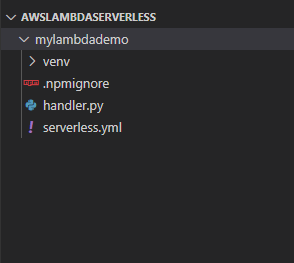
$ cd mylambdademo



You'll need to change into that directory and create a virtual environment for developing locally.

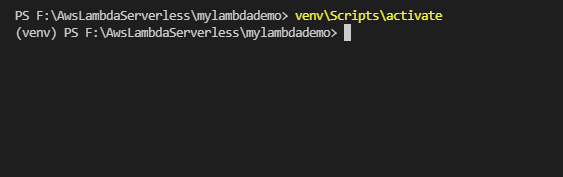
$ virtualenv venv --python=python3





Activate the virtual environment

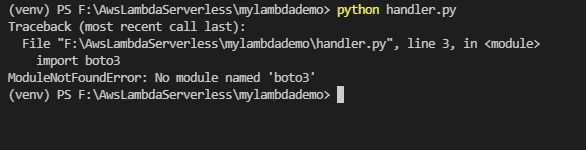
$ venv\Scripts\activate



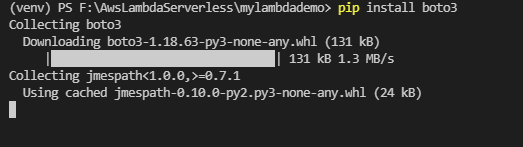
Let's set up the function we want to deploy. Edit the contents of handler.py

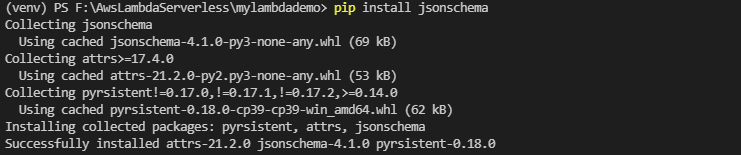
<https://github.com/vishnuvs369/Aws-Python/blob/main/validation/handler.py>

Run The handler now

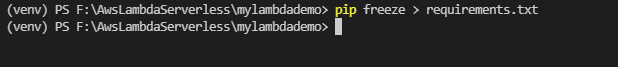


we haven't installed packages in our virtual environment yet. Let's do that now, and save the package versions of our environment to a requirements.txt file:





pip freeze > requirements.txt



Deploying Our function , and it's ready for us to deploy to Lambda. Edit the serverless.yml file to look like the following:

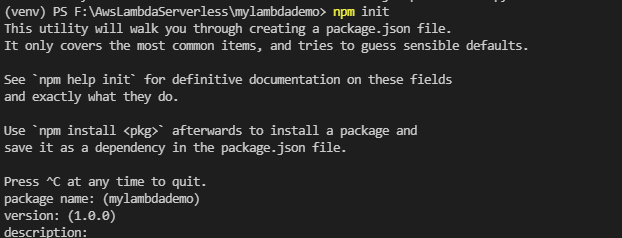
<https://github.com/vishnuvs369/Aws-Python/blob/main/validation/serverless.yml>

Change the handler name in severless.yml

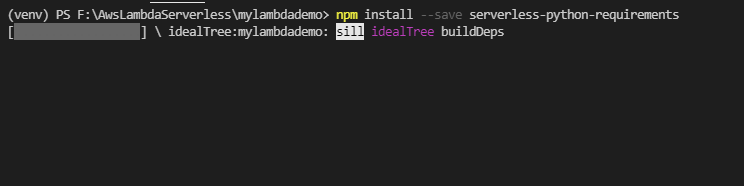
handler: handler.lambda\_handler

Our last step before deploying is to add the serverless-python-requirements plugin. Create a package.json file for saving your node dependencies. Accept the defaults, then install the plugin:

$ npm init



$ npm install --save serverless-python-requirements



To configure our serverless.yml file to use the plugin, we'll add the following lines in our serverless.yml:

plugins:

  - serverless-python-requirements

custom:

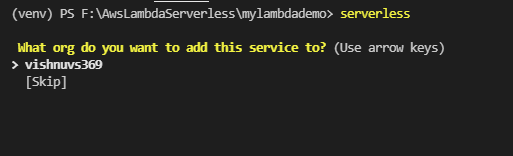
  pythonRequirements:

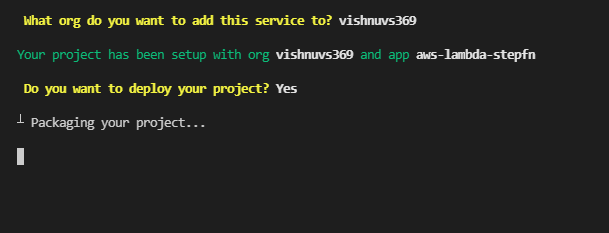
    dockerizePip: non-linux

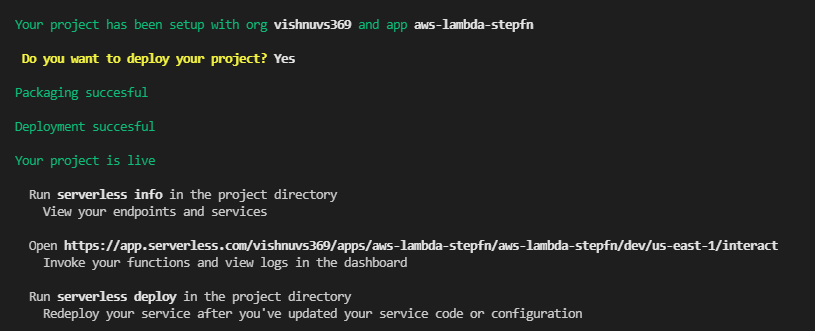
<https://github.com/vishnuvs369/Aws-Python/blob/main/validation/serverless.yml>

Now we can deploy our project

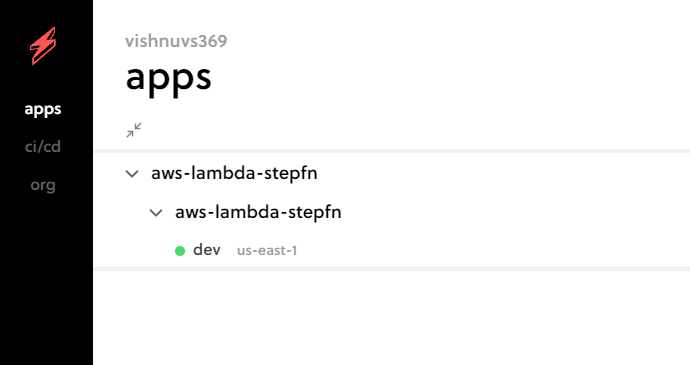
$ serverless



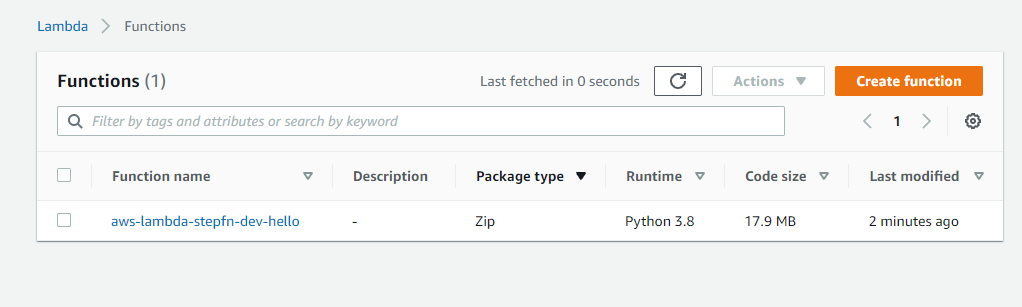


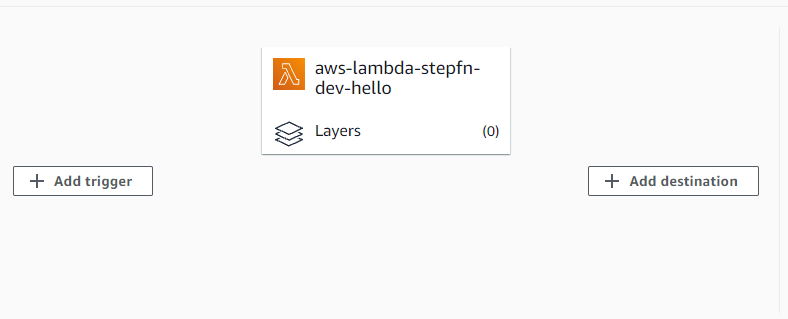


Go to serverless site where you can see your deployed project



And now check your AWS Lamba console it will be present there

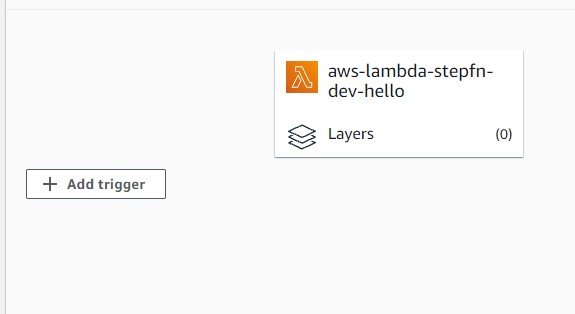


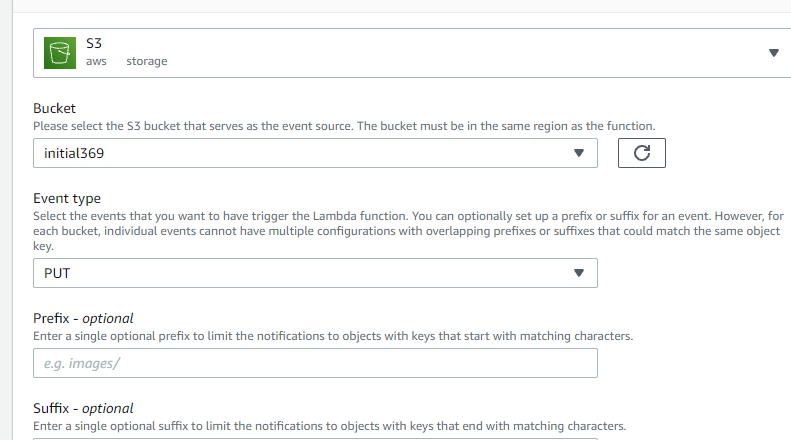


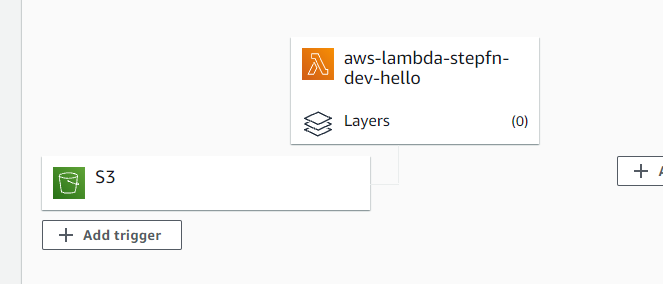
Now Create an AWS role that allows Aws S3 full Access, Aws Cloudwatch Full Access, Aws Stepfunction Full Acccess



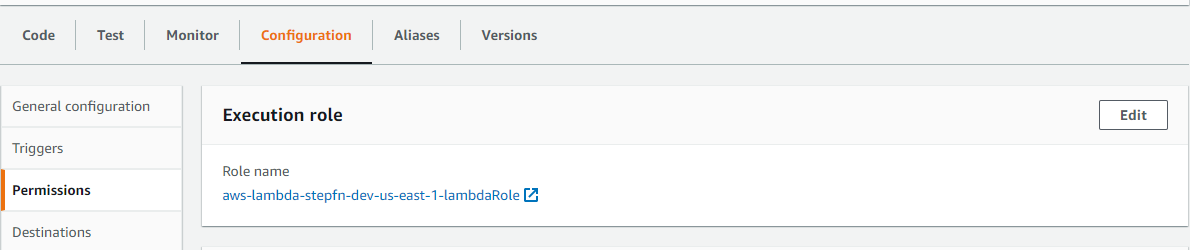
Now we can add S3 trigger to our lambda function



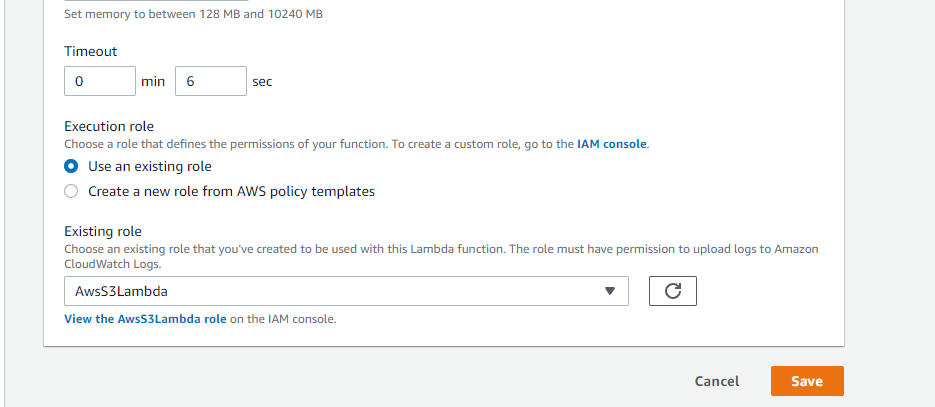




Now Update the role for our lamba function which gives full access to s3 and cloudwatch



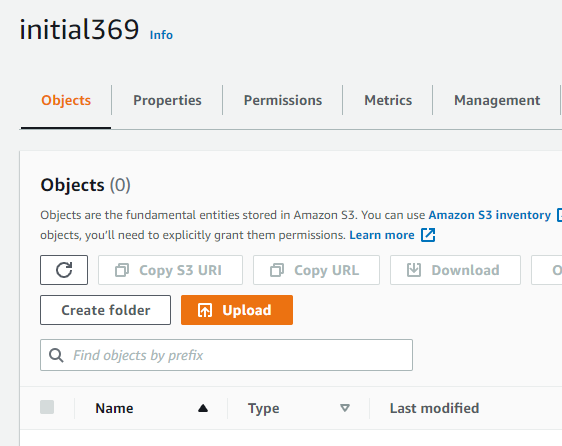
Give the previously created role to our lambda function

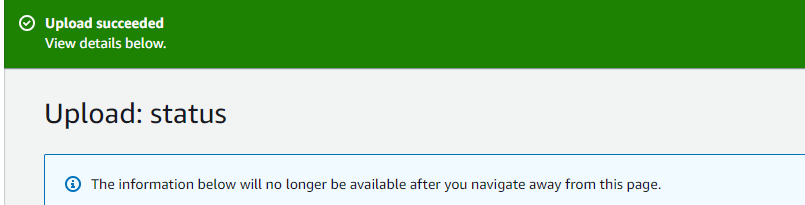


We have two buckets source bucket and destination bucket

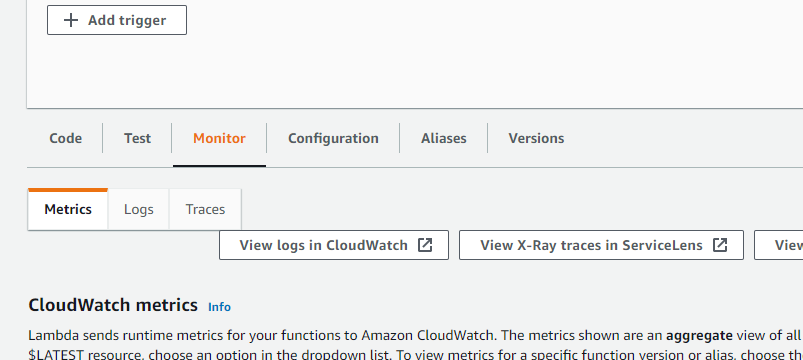
Now Add a demo json file in source bucket



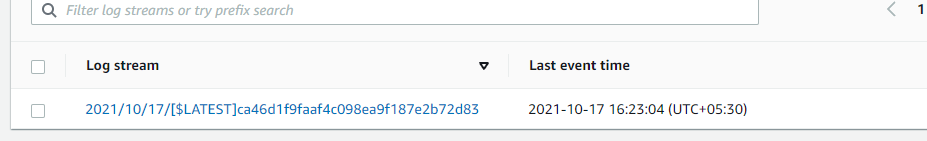




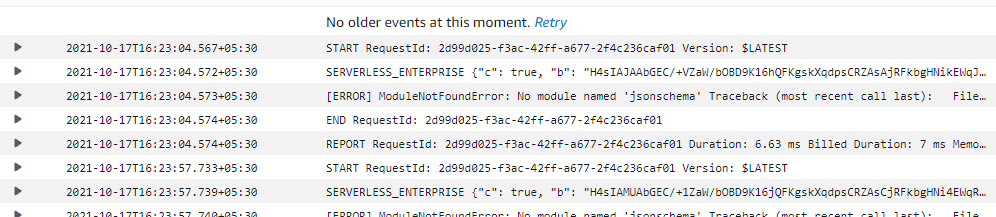
Now go to Aws Lambda Monitor and View Logs in CloudWatch



You can see one log present there open the log



You can see the logs there



For More reference

<https://www.serverless.com/blog/serverless-python-packaging>

<https://github.com/vishnuvs369/Aws-Python/blob/main/Docs/Serverless%20-%20Lab.pptx>

https://www.serverless.com/