# **CSYE 7245 - Big Data Sys and Int Analytics**

## **Lab 1 - Getting started with AWS + Lambda**

### **Team Information:**

|  |  |
| --- | --- |
| **Name** | **NUID** |
| Tanvi Gurav | 001306848 |
| Keerti Ojha | 001050173 |
| Priyanka Malpekar | 001302741 |

**Lab Completion Date: 22nd January’21**

# About

This lab demonstrated the implementation of AWS CLI and other services like IAM, Amazon S3, comprehend and lambda.



**Amazon Web Services (AWS)** provides computing resources and services that you can use to build applications within minutes at **pay-as-you-go pricing**. You can run nearly anything on AWS that you would run on physical hardware: websites, applications, databases, mobile apps, email campaigns, distributed data analysis, media storage, and private networks. The services they provide are designed to work together so that we can build complete solutions. There are currently dozens of services, with more being added each year. AWS is readily distinguished from other vendors because it is **flexible, cost-effective, scalable, elastic** and **secure**.



# Experiment setup

**Prerequisites:**

1. **Creating an AWS account**

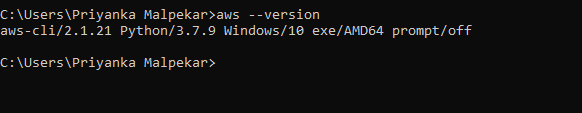
Creating an AWS account [here](https://aws.amazon.com/console/)

1. **Configuring AWS CLI**

Download AWS CLI from [here](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html)

**Configuring AWS on our system**

* Open cmd
* aws configure
* Enter Access Key and Secret Access Key
* Keep region and file format blank



1. **Starting with IAM ( Identity Access Management )**

* Always login as a root user to have all the privileges and access to all services
* Generate Access and Secret Access keys. Download it for future reference

1. **Creating a virtual environment**

Virtual environment helps us to create an isolated environment for all Python projects. Each project can thus have it’s own dependencies.

mkdir first-lambda

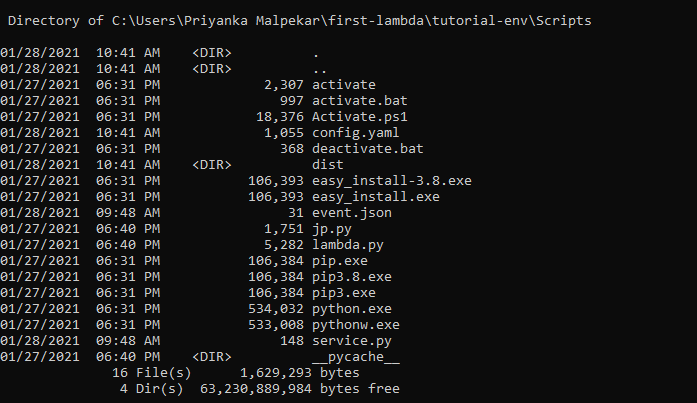
python3 -m venv tutorial-env

Activate the virtual environment → /Scripts/activate.bat

Install all the required packages in this virtual env - first-lambda

pip3 install Faker

pip3 install boto3

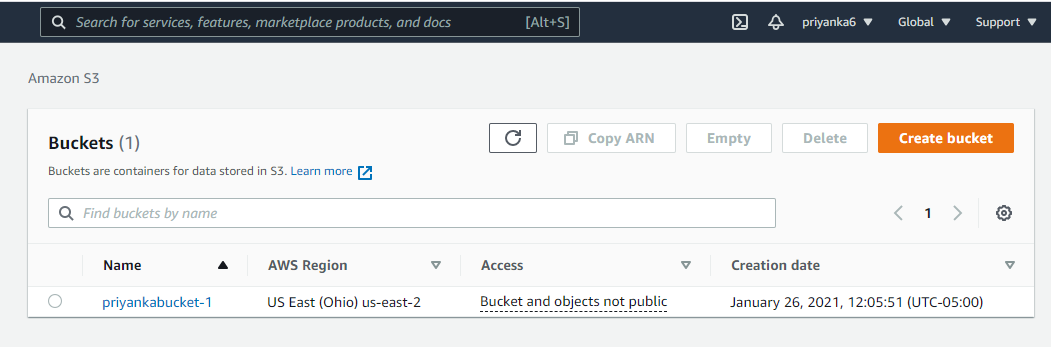


# Test Results

**Creating a S3 bucket**

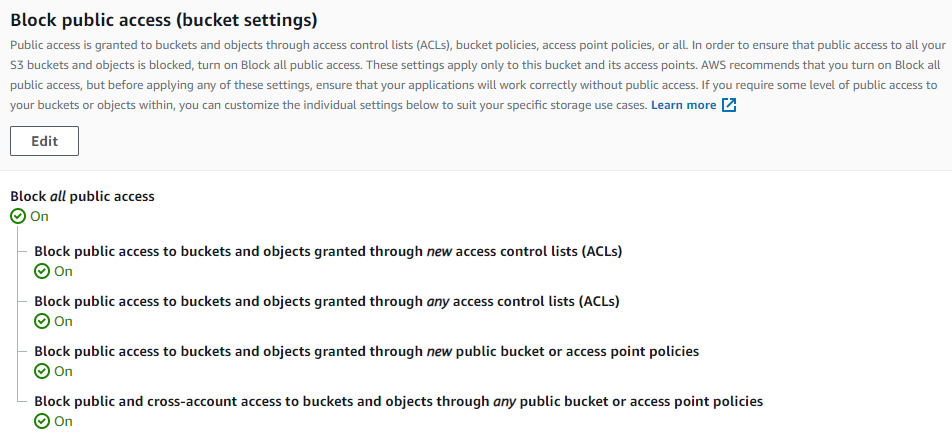


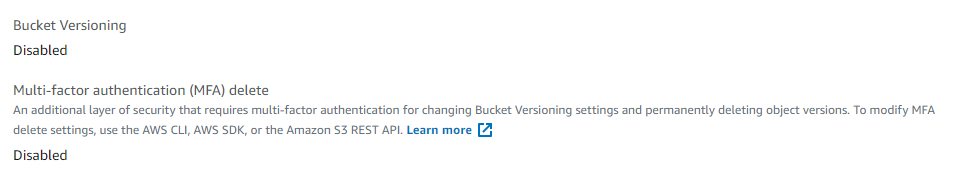
**Amazon S3** - **a simple storage service** is a scalable, high-speed, web-based cloud storage service. This service is designed for online backup and archiving of data and applications on Amazon Web Services (AWS).

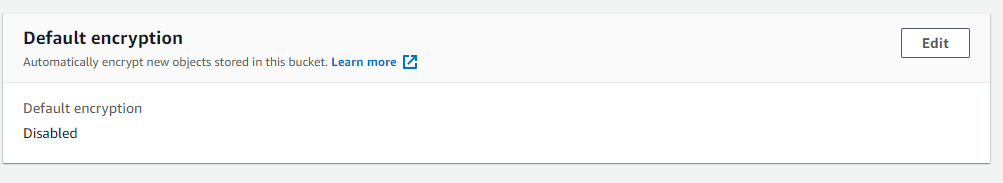


Ensure below mentioned rules while creating any S3 bucket:

* Block all public access
* Disable bucket versioning
* Disable encryption







# Use Cases

Let’s consider a simple use case to upload fake data in **S3 bucket** using Python packages like Faker and Boto3

**Faker**: A python package to generate fake data

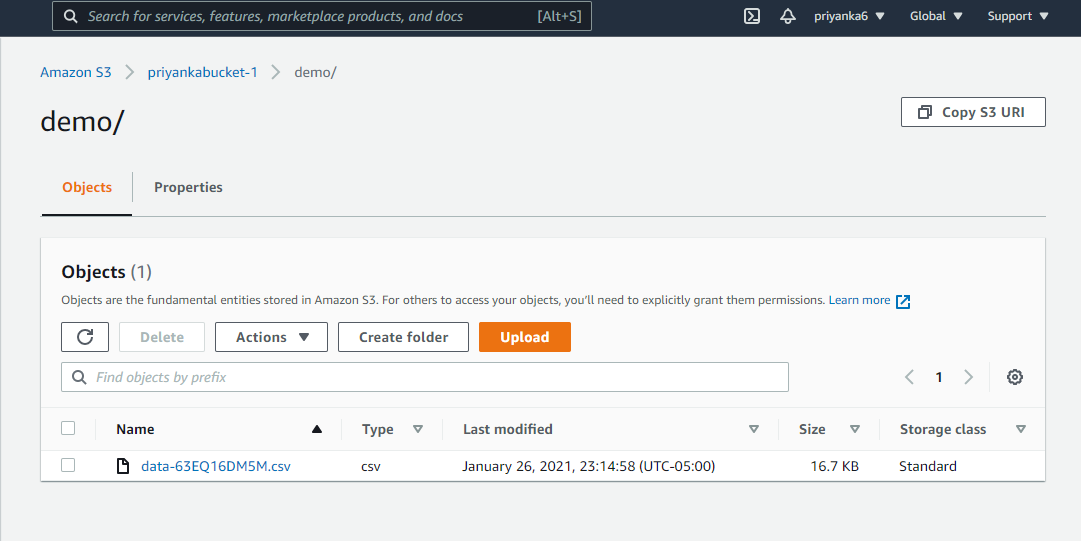
**Boto3**: Boto3 is a Amazon Web Services (AWS) Software Development Kit (SDK) for Python which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2

* **s3\_upload.py**

Python script to generate some fake data using Faker and upload to your S3 Bucket



S3 bucket **priyankabucket-1** created and fake csv uploaded in the respective bucket.

****

* **s3\_download.py**

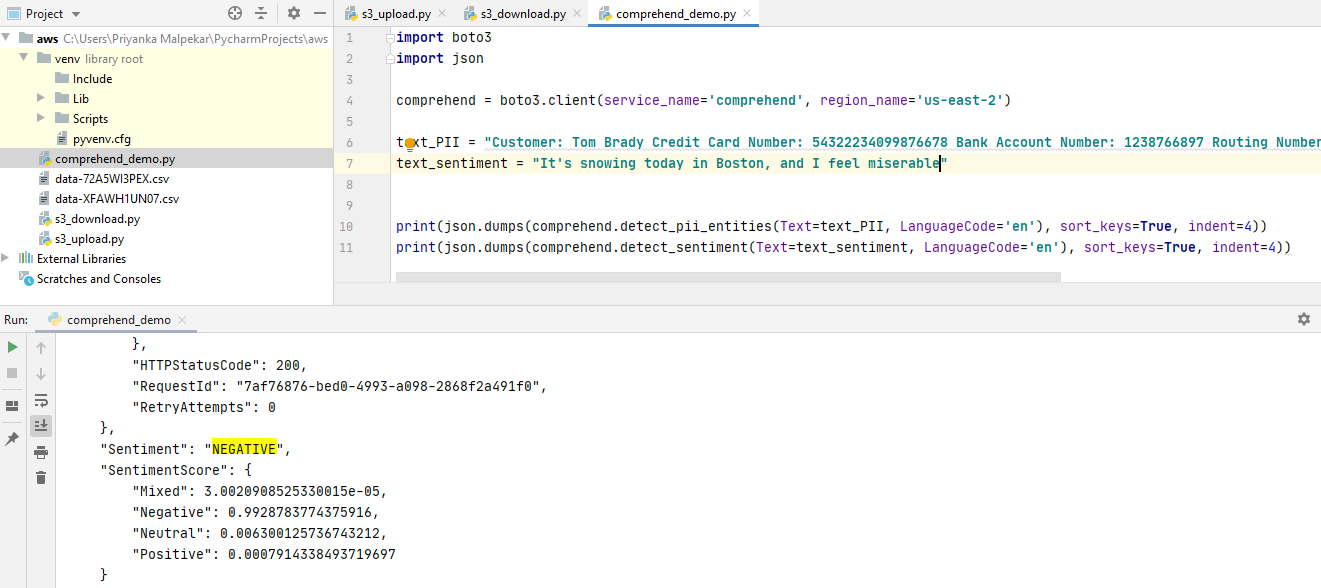
Download the generated csv file from S3 bucket to your local environment.

* **comprehend\_demo.py**

Using AWS **comprehend** service to detect sentiments. **Comprehend** is used for sentimental analysis. Consider the below mentioned use case to understand the concept of sentiment analysis using comprehend:

**miserable ---> Negative sentiment**

Every word has a sentiment score viz. **Mixed, Negative, Neutral** and **Positive**.

****

**happy ---> Positive sentiment**

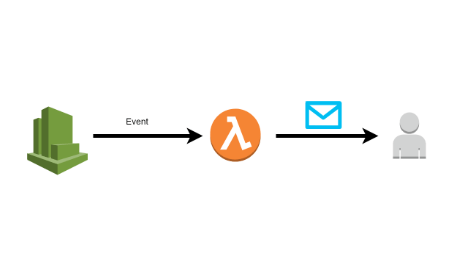
****

# 

# 

# 

# Lambda-serverless-py

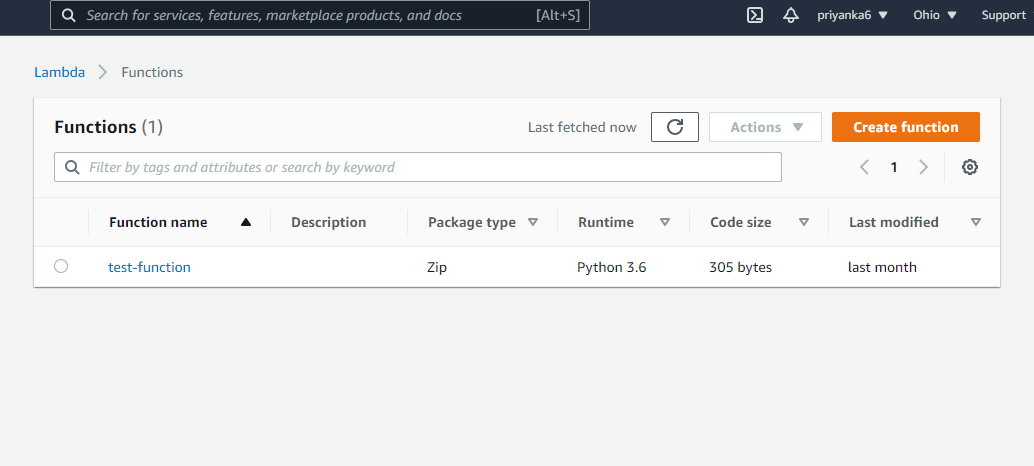


**AWS Lambda** is a **serverless** compute service that runs your code in response to events. It lets you run code without provisioning or managing servers. Lambda runs your code only when needed and scales automatically, from a few requests per day to thousands per second.

# 

**Creating a basic Lambda function**

# 

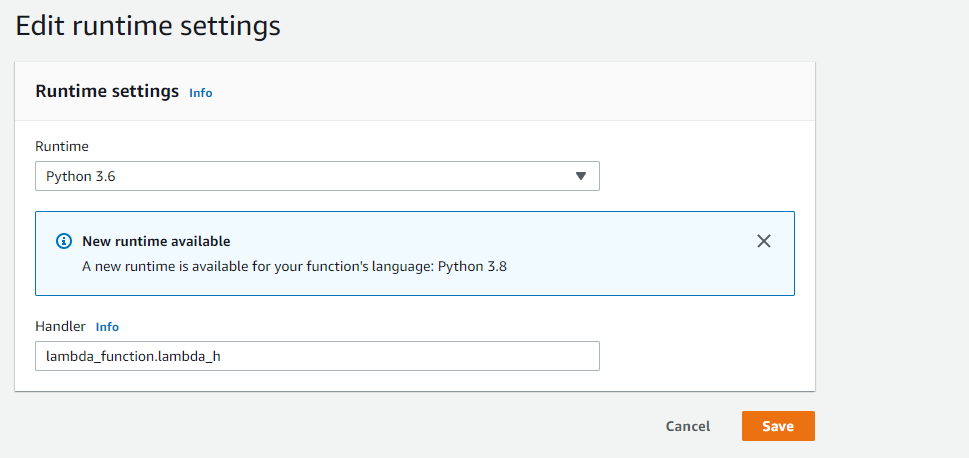


**Create an IAM Role**

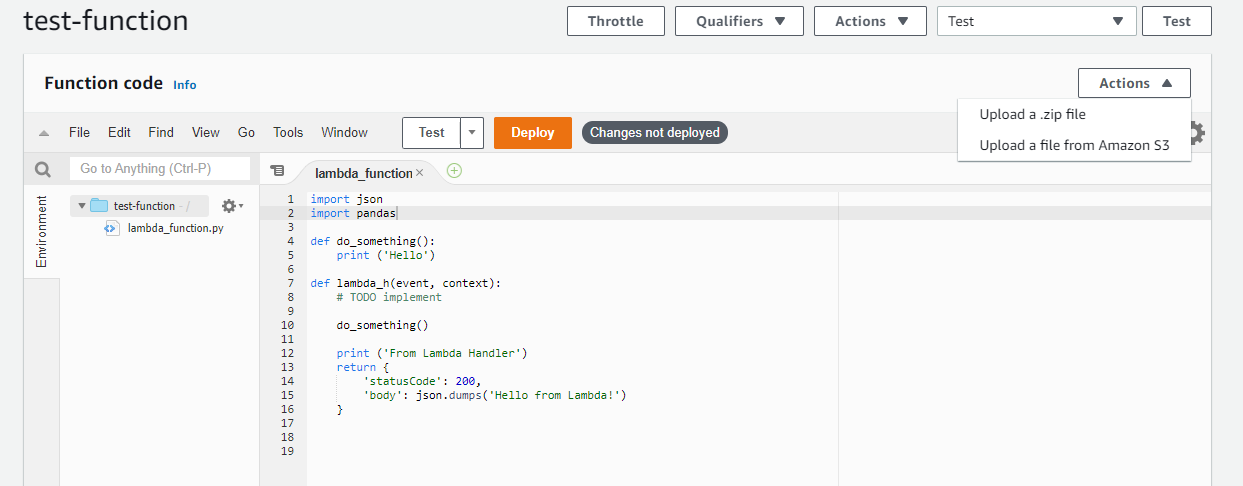
Creating a lambda role named ‘lambda\_basic\_execution’ with following privileges:

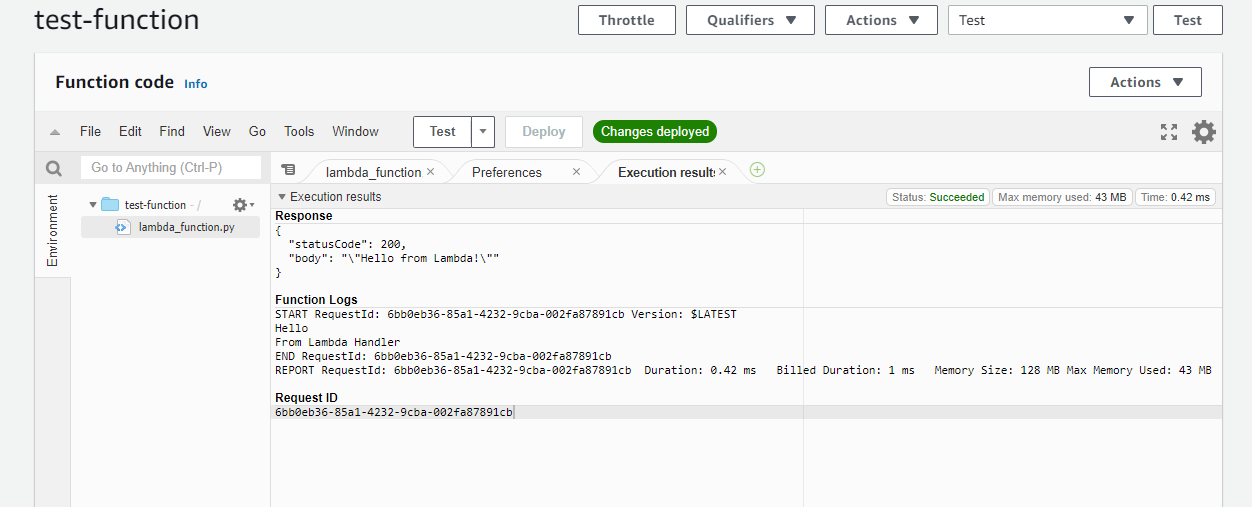
* Lambda basic execution
* Amazon S3 full access
* Amazon DynamoDB full access

Everything should be under **lambda\_handler** function . Rename it as lambda\_h by making changes in runtime settings.



**Executing the ‘test-lambda function’**





**Deploying Lambda function**

* Creating a virtual environment
* Installing required packages

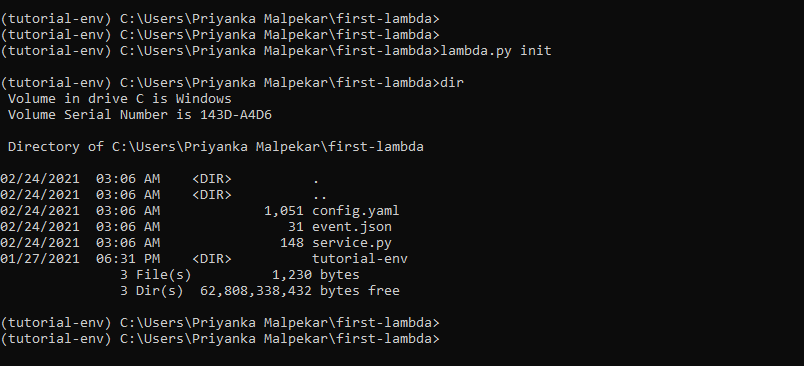
pip3 install python-lambda

pip3 install pandas

* Initiating lambda deployment

lambda.py init

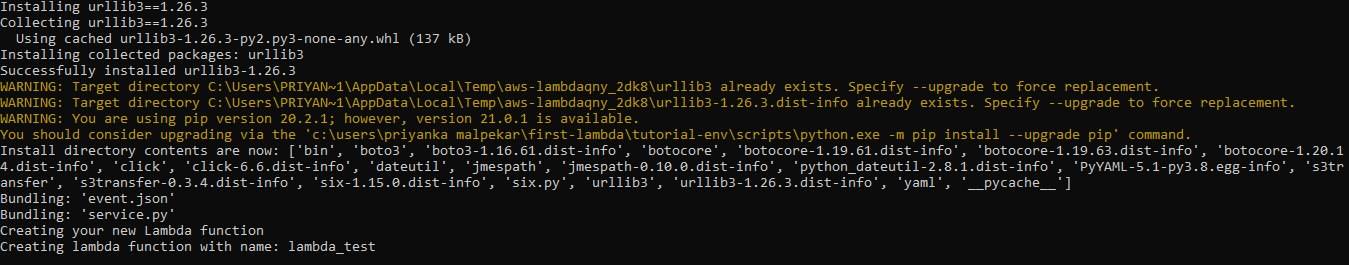
Three files will be generated viz. **Config.yaml, events.json, service.py**

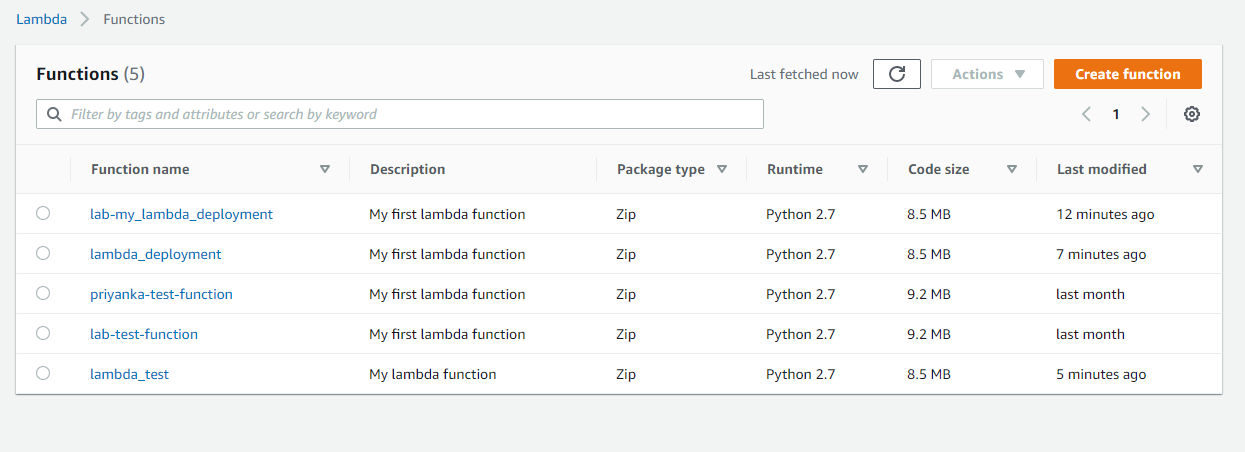


The **service.py** is the file we will be using. We can edit service.py with our Python code.

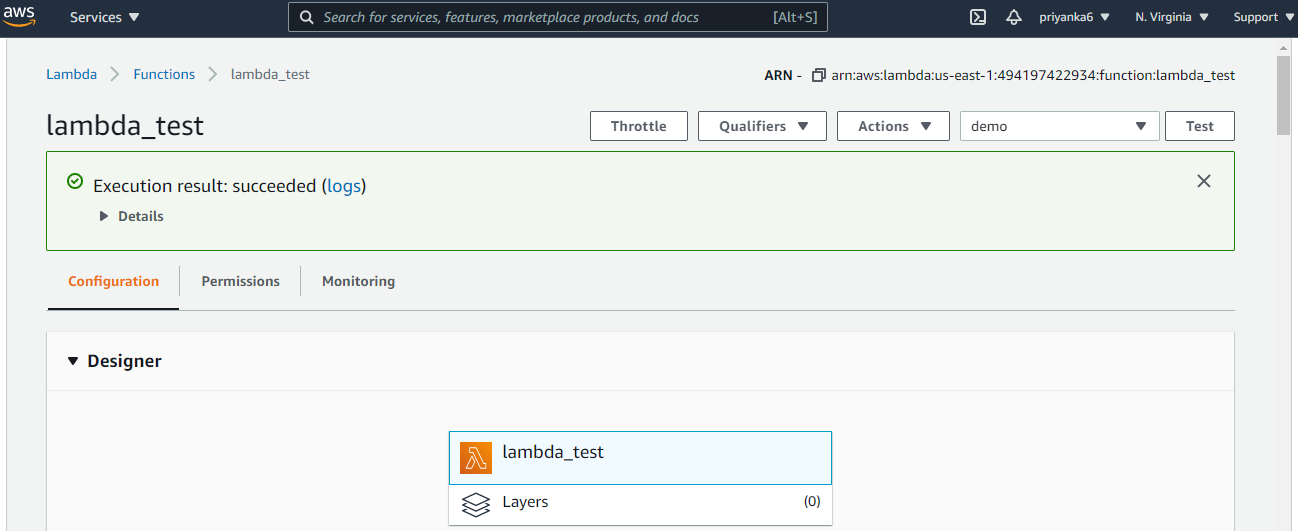
**Deploying lambda function**

lambda.py deploy





**Lambda function successfully deployed**



# 

# Lessons Learned

1. Uploading and downloading fake data using Faker in Amazon S3 bucket using python package like Boto3
2. Deploying lambda functions using AWS CLI
3. Learnt to use comprehend for sentiment analysis
4. Learnt about IAM roles and user permissions in AWS

# References

<https://media.amazonwebservices.com/AWS_Overview.pdf>

<https://awsdocs.s3.amazonaws.com/gettingstarted/latest/awsgsg-intro.pdf>

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>