Assignment Documentation on AWS Lambda Function- EC2 Instance

Ritesh

Dated: 31-12-2022

**What is AWS Lambda?**

**AWS Lambda** is a serverless, event-driven computing platform offered by Amazon as part of Amazon Web Services. It is a computing service that runs code in response to events and manages the computing resources required by that code automatically.

* [AWS Lambda](https://aws.amazon.com/lambda/) was created for use cases such as uploading images or objects to Amazon S3, updating DynamoDB tables, responding to website clicks, or responding to sensor readings from an IoT-connected device.
* AWS Lambda can also be used to automatically provision back-end services triggered by custom HTTP requests, as well as “spin down” such services when they are not in use to save resources.
* These custom HTTP requests are set up in AWS API Gateway, which can also handle authentication and authorization when used in conjunction with AWS Cognito.

### **Key Features of AWS Lambda**

Here are some features of AWS Lambda:

* **Integrated Fault Tolerance:**To help protect your code against individual machine or data center facility failures, AWS Lambda maintains compute capacity across multiple Availability Zones (AZs) in each AWS Region.
* **Scaling on the fly:**AWS Lambda executes your code only when it is required and scales automatically to meet the volume of incoming requests without any manual configuration. Your code has no limit on the number of requests it can handle.
* **Access Shared File Systems:**You can securely read, write, and persist large volumes of data at any scale using the Amazon Elastic File System (EFS) for AWS Lambda. To process data, you do not need to write code or download it to temporary storage. This saves you time, allowing you to concentrate on your business logic.

## Step 1: AWS Console → Lambda → Create Function.

## https://miro.medium.com/max/700/1*aJceEROiZufMvG7IpK6A1g.png

**Step 2: New tab → IAM → Roles → Pick the name of the role that you use created or used in Step 1.**

* Roles → pick the role name
* Under permissions, there is only one policy called “AWSLambdaBasicExecutionRole-##”. Select this role to add a custom execution policy to be able to create an EC2 instance.

## https://miro.medium.com/max/700/1*pHvJORyr-SQXYFO1stmDeA.png

## Select JSON

## https://miro.medium.com/max/700/1*BhXMzkhlgQY--HqyaQDjew.png

* Select Edit policy → delete the policy.
* Add the below-provided execution policy. This policy allows CloudWatch logs and it allows us to create log groups, create log streams, and put log events. The second part of the policy is for allowing us to create an EC2 instance.

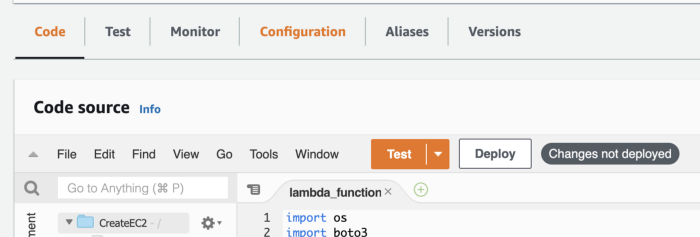
## 

**Step 3: Console → Lambda Management console**

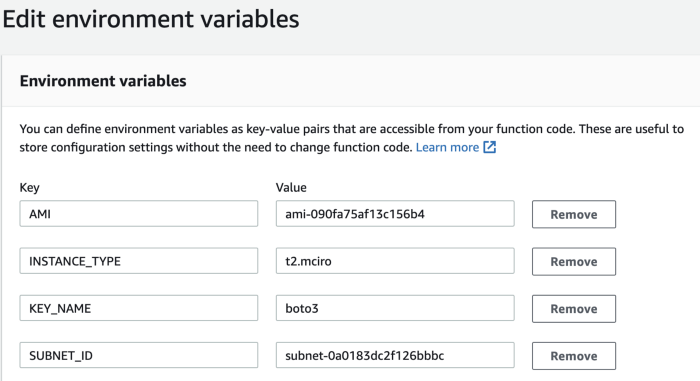
* Under lambda\_function, copy and paste the following code:
* Import OS: gives us access to our env variables
* Import boto3: importing boto3 library
* Starting from AMI → SUBNET\_ID, we are adding global EC2 variables. These values are not hardcoded because we are trying to create a reusable function.
* Defining entry point by using def lambda\_handler.

## 

## Step 4: Add environment variables



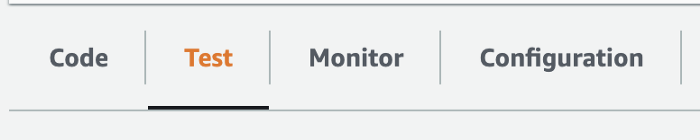
* Configuration → Environment Variable



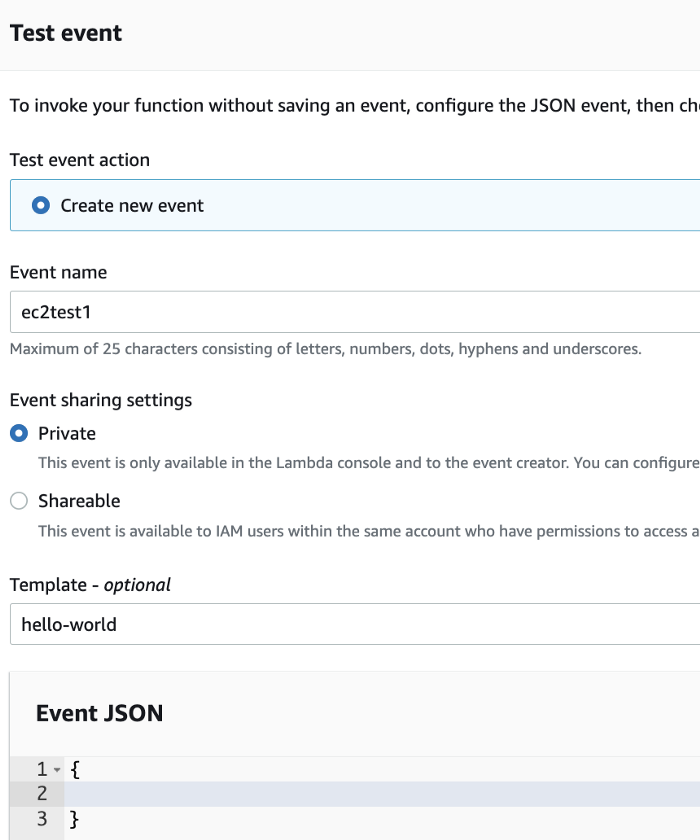
* Save

**Step 5: Run a test.**

* Test → test event



* Make sure it looks similar to this:



* Save it.
* Test it. You should see a successful execution.

