Lab 7: Run a Serverless "Hello, World!" with AWS Lambda

1. Open the lambda console.
2. In the AWS Lambda console, choose Create function.
3. Select use a blueprint.
4. In the Filter box, enter **hello-world-python** and select the **hello-world-python** blueprint.
5. Now enter Basic information about your Lambda function.
    a. Name: You can name your Lambda function here. **hello-world-python**
    b. Execution Role: Select **Create a new role from AWS policy templates**.
    c. Role name: type **lambda_basic_execution**.
6. Go to the bottom of the page and choose Create function.
7. Select Runtime: Currently, you can author your Lambda function code in Java, Node.js, C#, Go, or Python. For this case, use **Python 3.7** as the runtime.
8. Handler: You can specify a handler (a method/function in your code) where AWS Lambda can begin executing your code. AWS Lambda provides event data as input to this handler, which processes the event. **lambda_function.lambda_handler**
9. Select **Configure Test Event** from the drop-down menu called **Test**.
10. The editor pops up so you can enter an event to test your function.
    a. Select Create new event.
    b. Type in an event name like **HelloWorldEvent**.
    c. Retain default setting of **Private** for Event sharing settings.
    d. Choose **hello-world** from the template list.
    e. You can change the values in the sample JSON, but don't change the event structure. For this, replace value1 with hello, world!.
    f. Select save.
11. Choose Test.
12. Upon successful execution, view the results in the console:
    a. The Execution results tab verifies that the execution succeeded.
    b. The Function Logs section will show the logs generated by the Lambda function execution as well as key information reported in the Log output.
13. **Monitor your metrics**: AWS Lambda automatically monitors Lambda functions and reports metrics through Amazon CloudWatch. To help you monitor your code as it executes, Lambda automatically tracks the number of requests, the latency per request, and the number of requests resulting in an error and publishes the associated metrics.
14. Invoke the Lambda function a few more times by repeatedly choosing the Test button. This will generate the metrics.

Lab: Create a lambda function to start and stop ec2 instance.

Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume—there is no charge when your code is not running (and you start paying only after the first million requests per month on the AWS Free Tier). With Lambda, you can run code for virtually any type of application or backend service—all with zero administration. Upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically be initiated from other AWS services or events, or it can be set up to respond directly to an HTTP or HTTPS request.

One major difference between Lambda and Amazon EC2 is the cost of operation. The pricing models of the two services are quite different. Lambda cost is based on usage of the Lambda functions and the amount of storage used by the functions, whereas Amazon EC2 cost is based on the type of machine image used and the amount and type of storage used. For light and medium workloads, Lambda is a dramatically less expensive solution than Amazon EC2.

An inline policy is a policy that's embedded in an IAM identity (a user, group, or role). That is, the policy is an inherent part of the identity. You can create a policy and embed it in an identity, either when you create the identity or later.

1. Create and Ec2 instance
2. Create a new role.
    a. Under the use case select lambda in place of Ec2. Don’t select or create any policy as we create the role with inline policy after creation of the role.
    b. Click on the inline policy.
    c. Choose a service. Select Ec2.
    d. In the actions, select startinstances and stopinstances.
    e. Under the resources, choose specific resource.
    f. Click on the Add ARN. Add your region, account id and instance id. Now click on the Add.
    g. Now click on the review policy and create policy.
3. Now create the new lambda function and run from scratch.
    a. Give some name to function. Select the runtime environment code.
    b. Under the permission, select the existing role and choose the role created in the step 2.
    c. Now click on the create function.
    d. Now in the code window write the following code:

```
import json
import boto3
region='us-east-1'
instances=['i-0d3d873ddbf0b5fb7']
def lambda_handler(event,context):
    ec2=boto3.client('ec2',region_name=region)
    ec2.start_instances(InstanceIds=instances)
    print('your instance has been stopped'+str(instances))
```

    e. Configure the running time of lambda function by clicking on the configuration tab. Through edit, set it as 10 second in place of default 3 second value.
    f. Save the python file and deploy the changes and test the code. If instance is running then change the last two lines of the code accordingly and vice versa.