

Karthik Pradeep Hegadi

2KE20CS032

Assignment 48

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

AWS

Assignment: 1 : Serverless computing using AWS LAMBDA

Overview:

1. Deploy code in AWS lambda
2. Create and invoke API for lambda function

Code deployment using AWS Lambda

1. Navigate to Lambda page and click on create function
2. Provide the function name
3. Select the Runtime as python 3.9
4. Click on create function and it should be created successfully

Lambda > Functions > Create function

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from a blueprint.

Basic information

Function name
Enter a name that describes the purpose of your function.

lamdafunction

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding permissions.

[Change default execution role](#)

5. Scroll down and you can see a sample python code that you can use it for the deployment
6. Go to test tab to create a new event, provide the event name and click on save event and the event will get saved
7. Now navigate back to the code tab and click on test

The screenshot shows the 'Test' tab in the AWS Lambda console. At the top, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Test' tab is active. Below the tabs, there's a 'Test event' section with an 'Info' link. A message states: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', the 'Create new event' option is selected. The 'Event name' field contains 'landafuncion-event01', with a note below it: 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.' Under 'Event sharing settings', the 'Private' option is selected, with a note: 'This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)'. The 'Shareable' option is also visible with a note: 'This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)'. Under 'Template - optional', the text 'hello-world' is entered. At the bottom, the 'Event JSON' is shown as a code block with the following content:

```
1 - {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

8. You will get the execution results as below

The screenshot shows the 'Code source' tab in the AWS Lambda console. At the top, there are tabs for 'Code source', 'Info', 'Test', and 'Deploy'. The 'Code source' tab is active. Below the tabs, there's a search bar 'Go to Anything (% P)' and a list of files in the 'Environment' section: 'lamdafuncion' and 'lambda_function.py'. The 'Execution results' section is expanded, showing the following information:

- Test Event Name:** landafuncion-event01
- Response:**

```
{  
  "statusCode": 200,  
  "body": "\"Hello from Lambda!\""  
}
```
- Function Logs:**

```
START RequestId: 1c6b779d-4aff-4a49-b7e4-b94ddb63d8d Version: $LATEST  
END RequestId: 1c6b779d-4aff-4a49-b7e4-b94ddb63d8d  
REPORT RequestId: 1c6b779d-4aff-4a49-b7e4-b94ddb63d8d  Duration: 1.42 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory
```
- Request ID:** 1c6b779d-4aff-4a49-b7e4-b94ddb63d8d

Configure REST API to call the lambda function

1. Navigate to API gateway dashboard
2. Scroll down and select Build option on REST api in the API type
3. Click on New api, provide the api name, description and click on create API you will get a response as below



API Gateway > APIs > Create API > Create REST API

Create REST API

API details

☒ **New API**
Create a new REST API.

☐ **Clone existing API**
Create a copy of an API in this AWS account.

☐ **Import API**
Import an API from an OpenAPI definition.

☐ **Example API**
Learn about API Gateway with an example API.

API name
my-reset-api-01

Description - optional
this api is created for assginment of gradious

API endpoint type
Regional

Cancel Create API

Delete

Create method

Method	Integration type	Authorization	API key
No methods			
No methods defined.			

☐ Lambda proxy integration
 Send the request to your Lambda function as a structured event.

Lambda function
 Provide the Lambda function name or alias. You can also provide an ARN from another account.

ap-south-1

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.


4. Navigate to Actions and click on 'Create Method' and select GET from the resource dropdown and click on tick mark


Method type


GET

A method type is required.

Integration type

☒ Lambda function
 Integrate your API with a Lambda function.
 

☐ HTTP
 Integrate with an existing HTTP endpoint.
 

☐ Mock
 Generate a response based on API Gateway mappings and transformations.
 

5. In the LAMBDA function text box type the lambda function that you have created before and leave remaining all settings as default

6. Click on Save and then click 'OK' in the prompt

7. Resource will be created and you will get a response as below

8. Click on Actions and click on Deploy API

API actions ▼

Deploy API

Update documentation

Delete

4m367xosn8/*/GET/	Resource ID fe67t764rb
-------------------	---------------------------

9. Select the Deployment stage as New stage, provide the stage name, stage description, and deployment description and click on deploy

[API Gateway](#) > [APIs](#) > [my-reset-api-01 \(4m367xosn8\)](#) > [Stages](#)

Stages

stage-01

Stage details

Info

Stage name

stage-01

Rate

Info

-

API cache

Inactive

Burst

Info

-

Invoke URL

https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-01

Active deployment

10. You will get a response page as below

11. Now click on the Invoke url and you will get the function invoked using REST API

Stage name stage-01	Rate Info -
API cache ⊖ Inactive	Burst Info -
Invoke URL https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-01	
Active deployment m689m7 on December 31, 2023, 10:49 (UTC+05:30)	
Logs and tracing Info	
CloudWatch logs ⊖ Inactive	Detailed metrics ⊖ Inactive

← → ↺

https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-0

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

statusCode: 200

body: "Hello from Lambda!"