

# How to create Rest API using Lambda & ApiGateway

## Rest API using Lambda & ApiGateway - Created By Lasantha

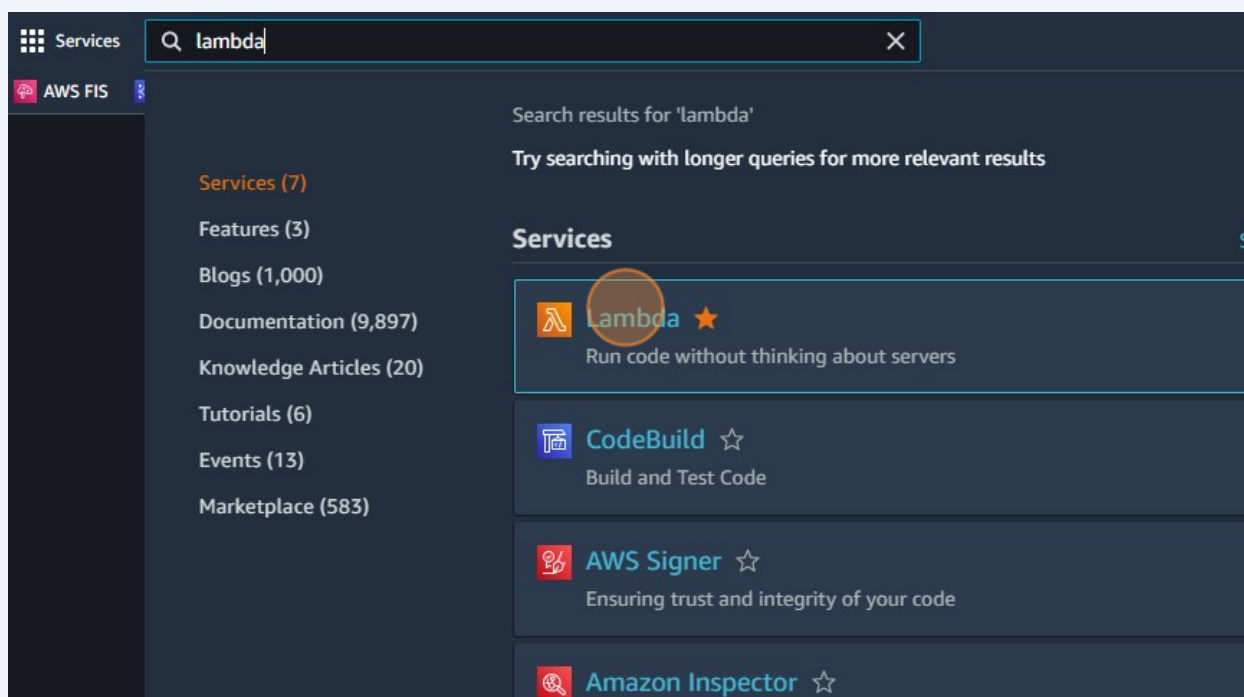
1

Navigate to

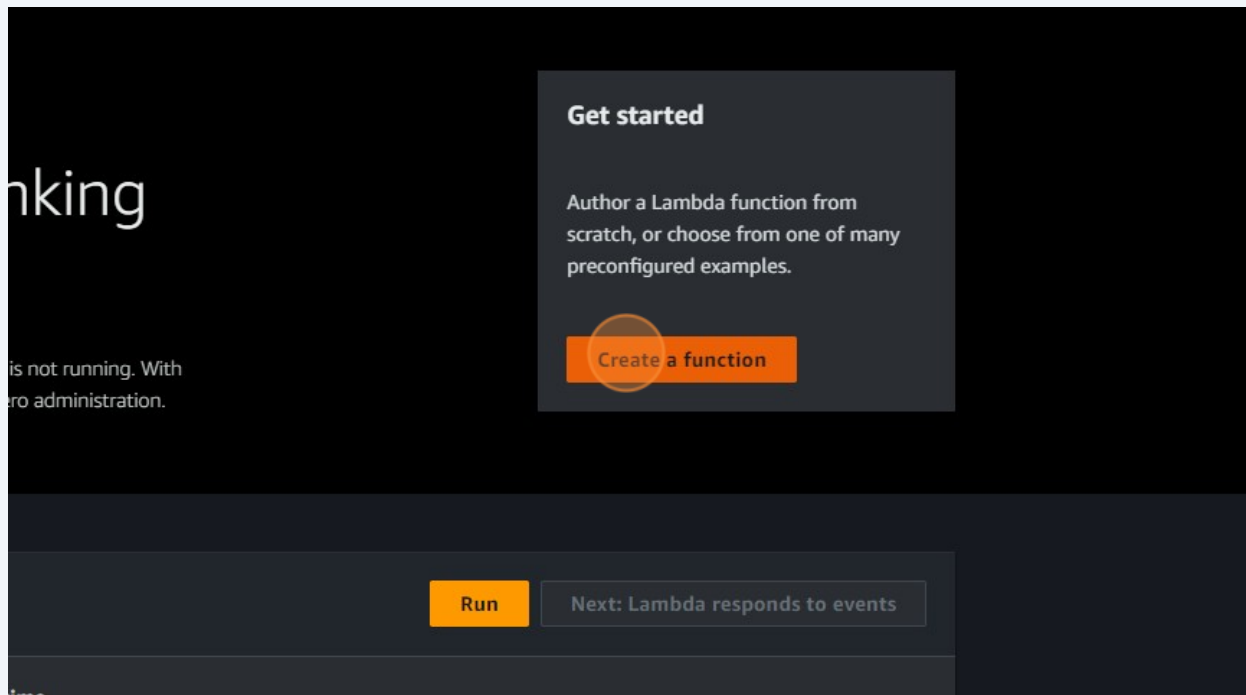
<https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1>

2

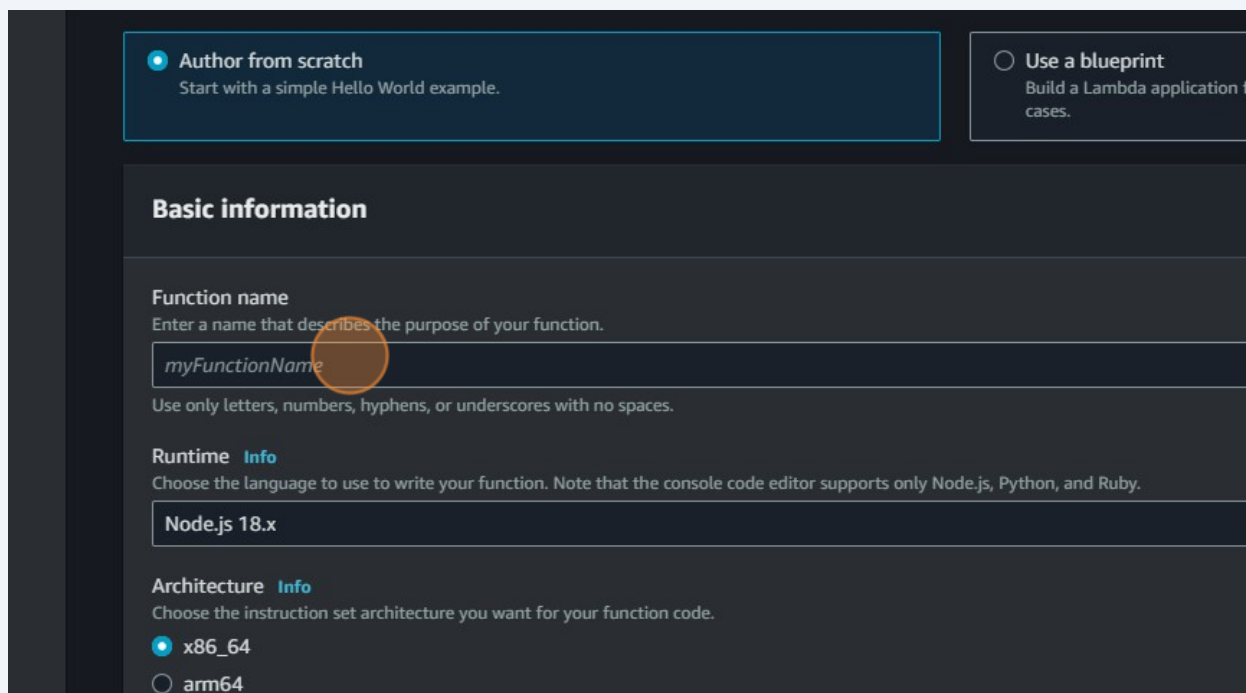
Click "Lambda"



3 Click "Create a function"

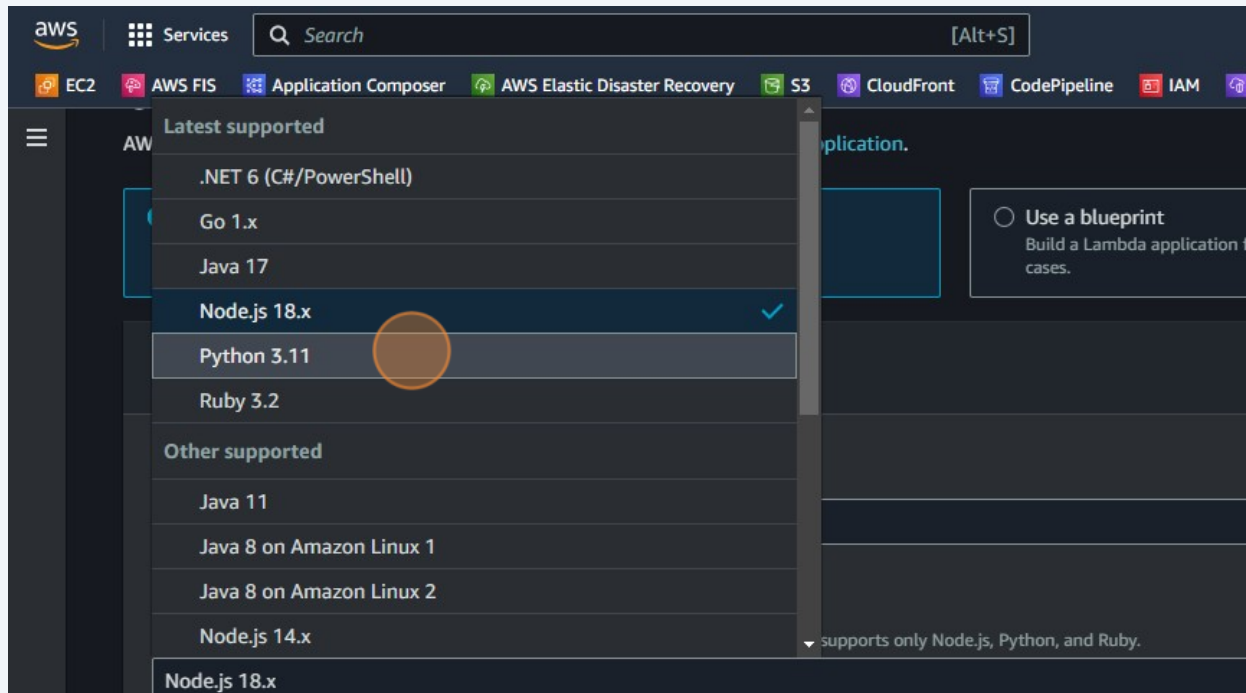


4 Click the "Function name" field.

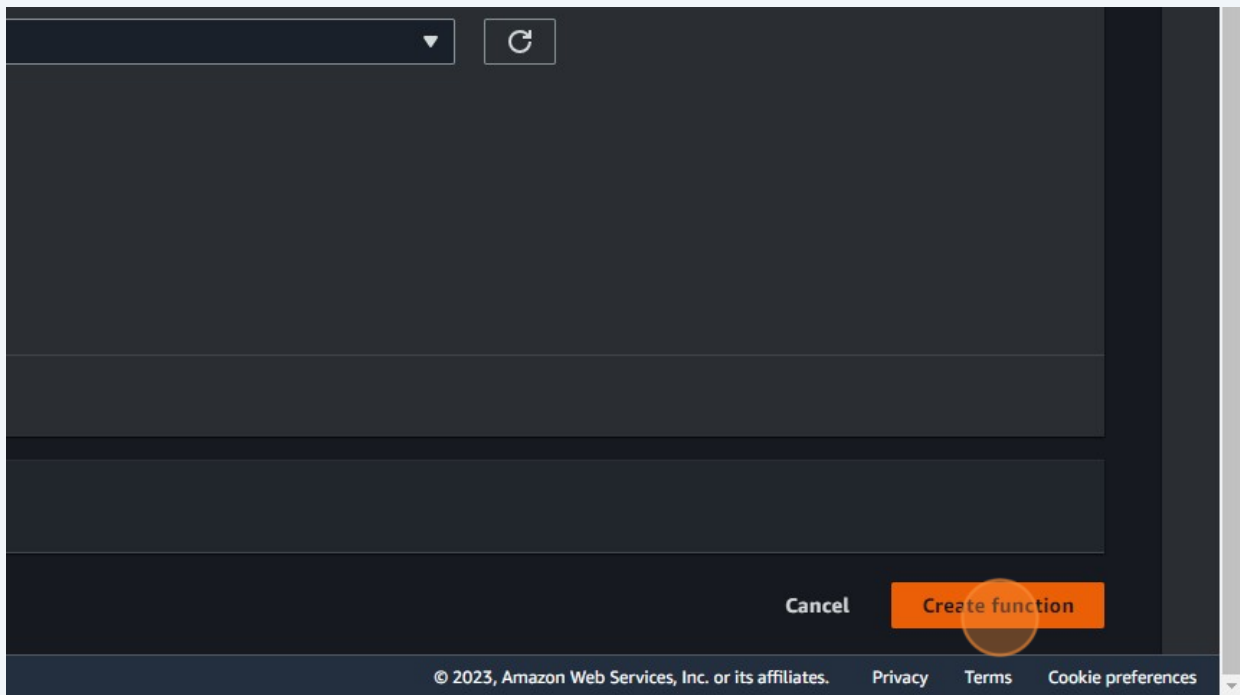


5 Type "aws-sinhalen-app-fun"

6 Click "Python 3.11"

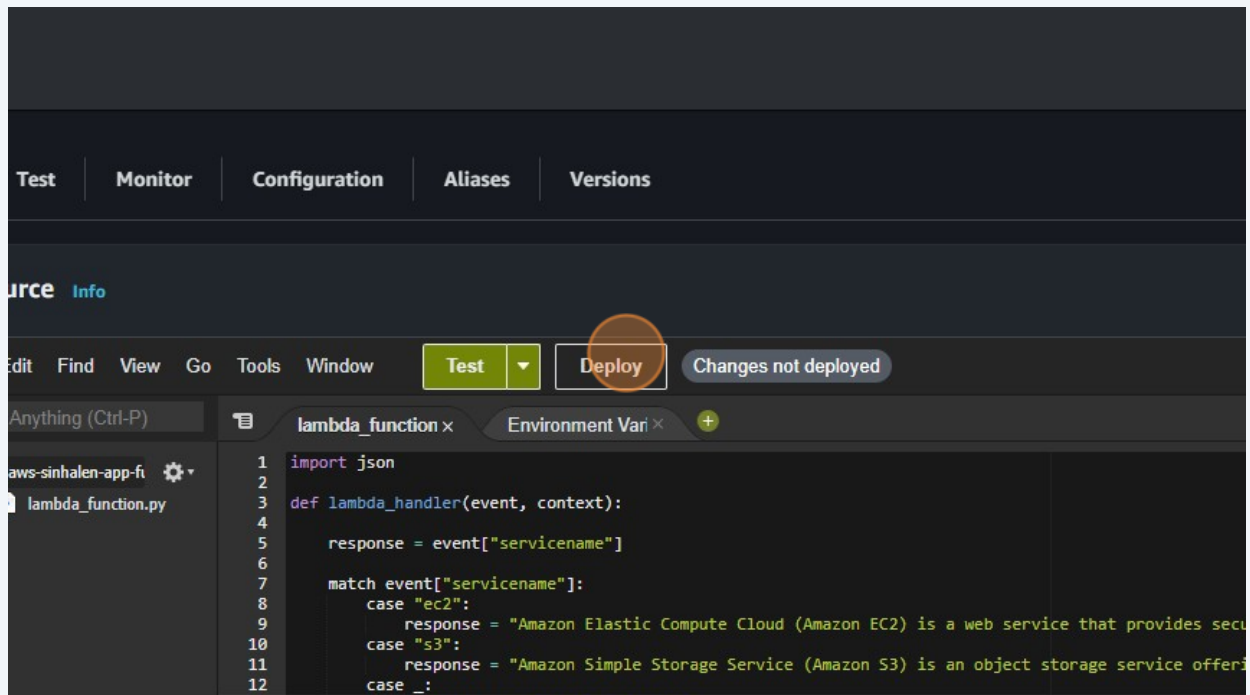


7 Click "Create function"

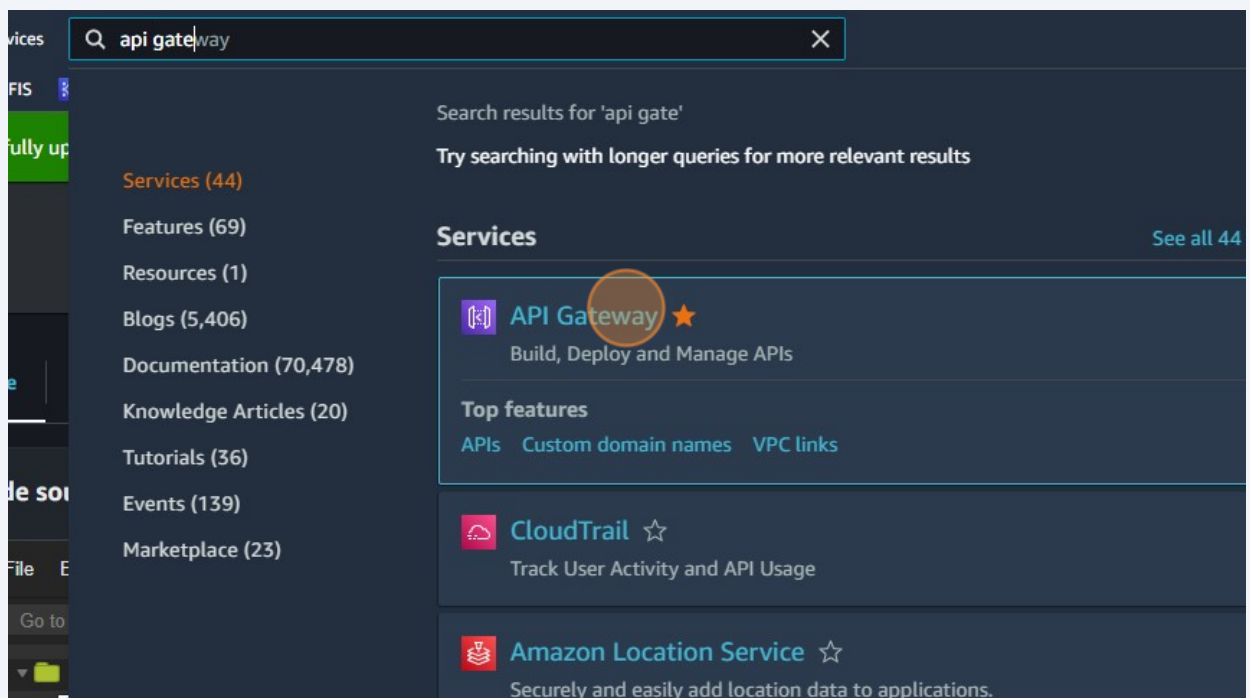


8 In the code section add following code  
<https://gist.github.com/sanju2/0ba05a7c005f0f5d22066135b1b1bb55>

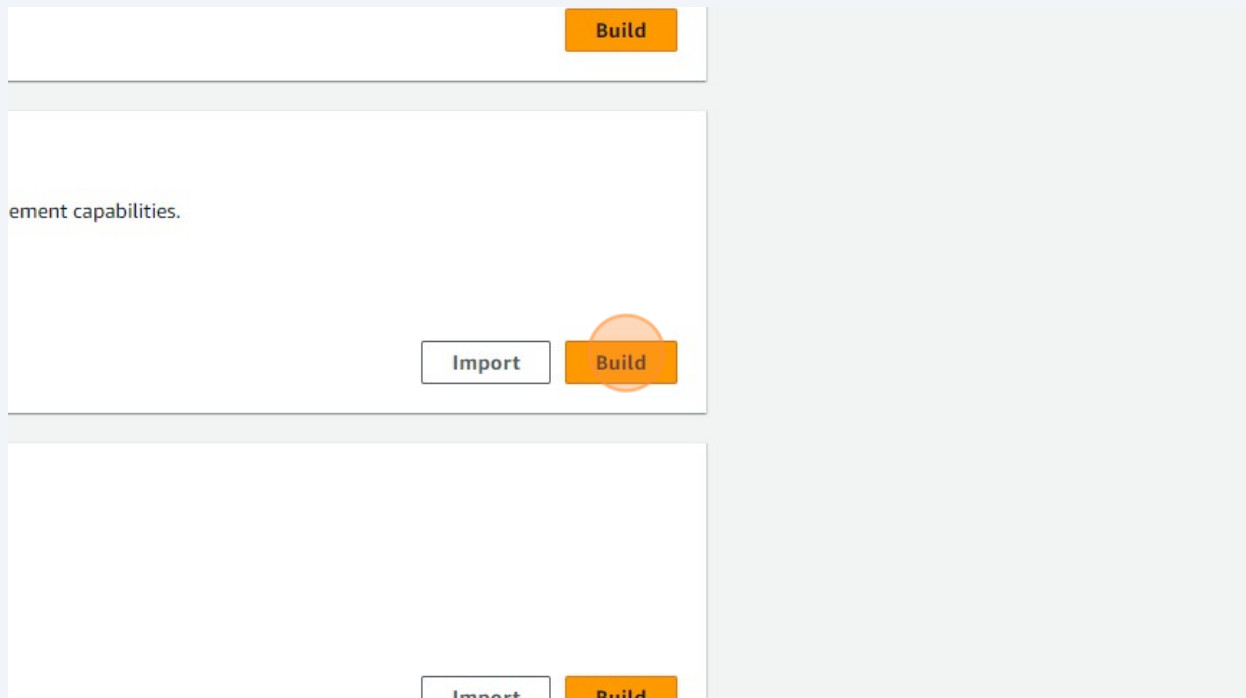
## 9 Click "Deploy"



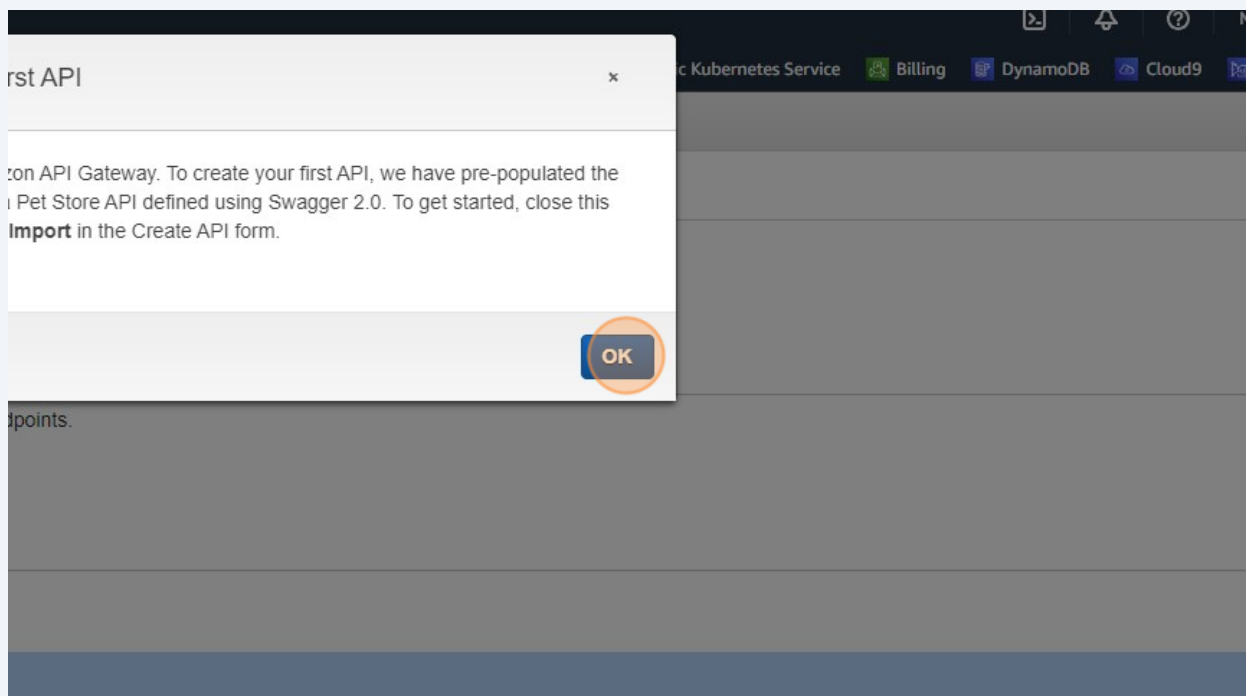
## 10 Right-click "API Gateway"



## 11 Click "Build"



## 12 Click "OK"



### 13 Click "New API"

#### Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

☒ REST ☐ WebSocket

#### Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☐ New API ☐ Import from Swagger or Open API 3 ☒ Example API

#### Example API

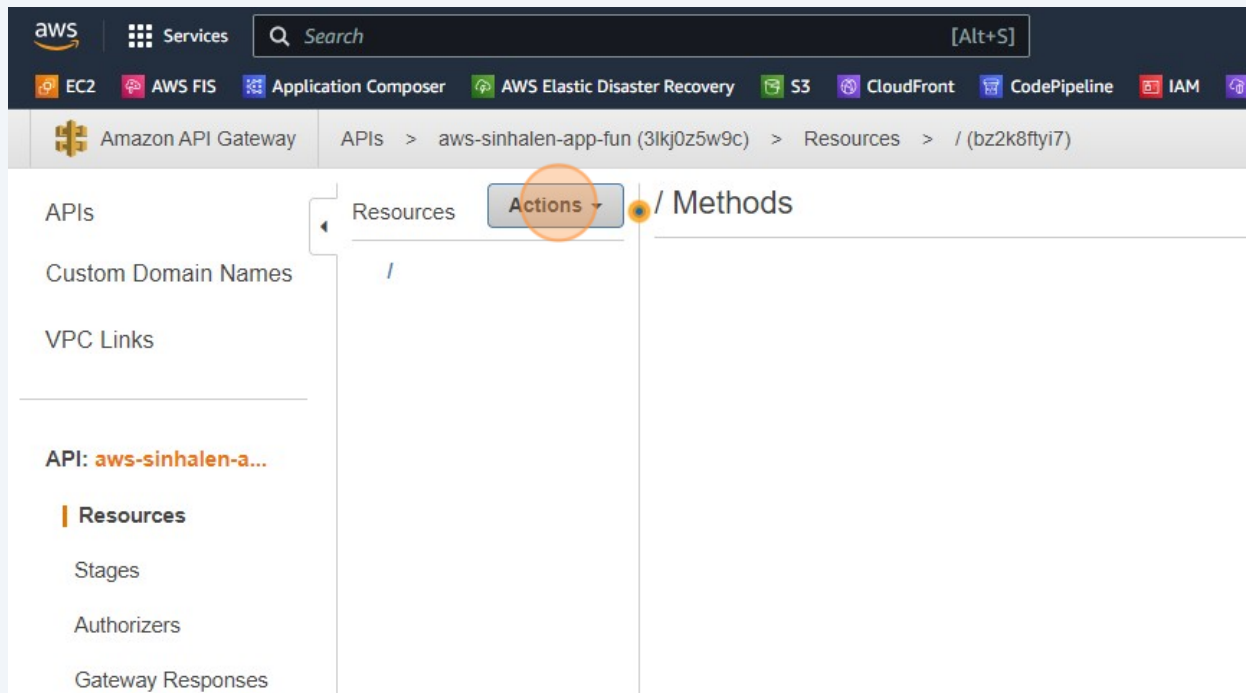
Learn about the service by importing an example API and turning on hints throughout the console.

```
1 {  
2   "swagger": "2.0",  
3   "info": {  
4     "description": "Your first API with Amazon API Gateway. This is a sample API that integrates via HTTP",  
5     "title": "PetStore"  
6   },  
7   "schemes": [  
8     "https"  
9   ],  
10  "paths": {  
11    "/pets": {  
12      "get": {  
13        "summary": "List all pets",  
14        "operationId": "listPets",  
15        "consumes": ["application/json"],  
16        "produces": ["application/json"],  
17        "responses": {  
18          "200": {  
19            "description": "A list of pets",  
20            "schema": {  
21              "type": "array",  
22              "items": {  
23                "$ref": "#/definitions/pet"  
24              }  
25            }  
26          }  
27        }  
28      }  
29    }  
30  }  
31  "definitions": {  
32    "pet": {  
33      "type": "object",  
34      "required": ["id", "name"],  
35      "properties": {  
36        "id": {  
37          "type": "integer",  
38          "format": "int64"  
39        },  
40        "name": {  
41          "type": "string"  
42        },  
43        "tag": {  
44          "type": "string"  
45        }  
46      }  
47    }  
48  }  
49 }
```

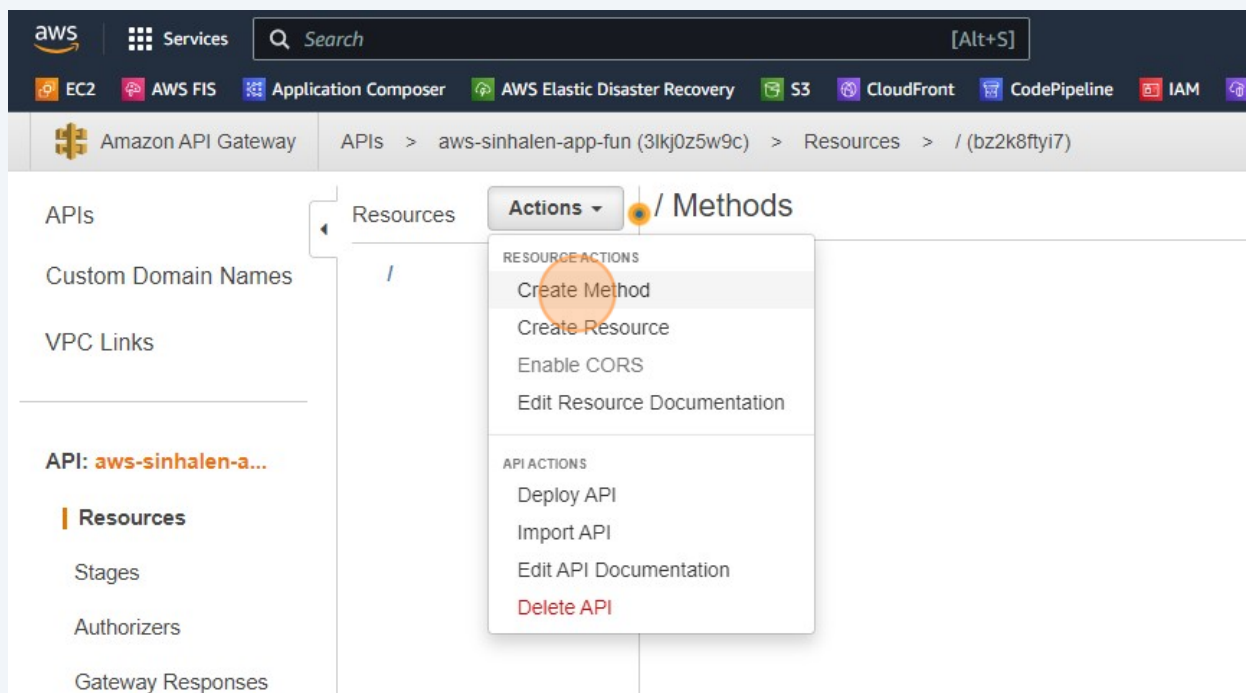
### 14 Click "Create API"

Create API

## 15 Click "Actions"

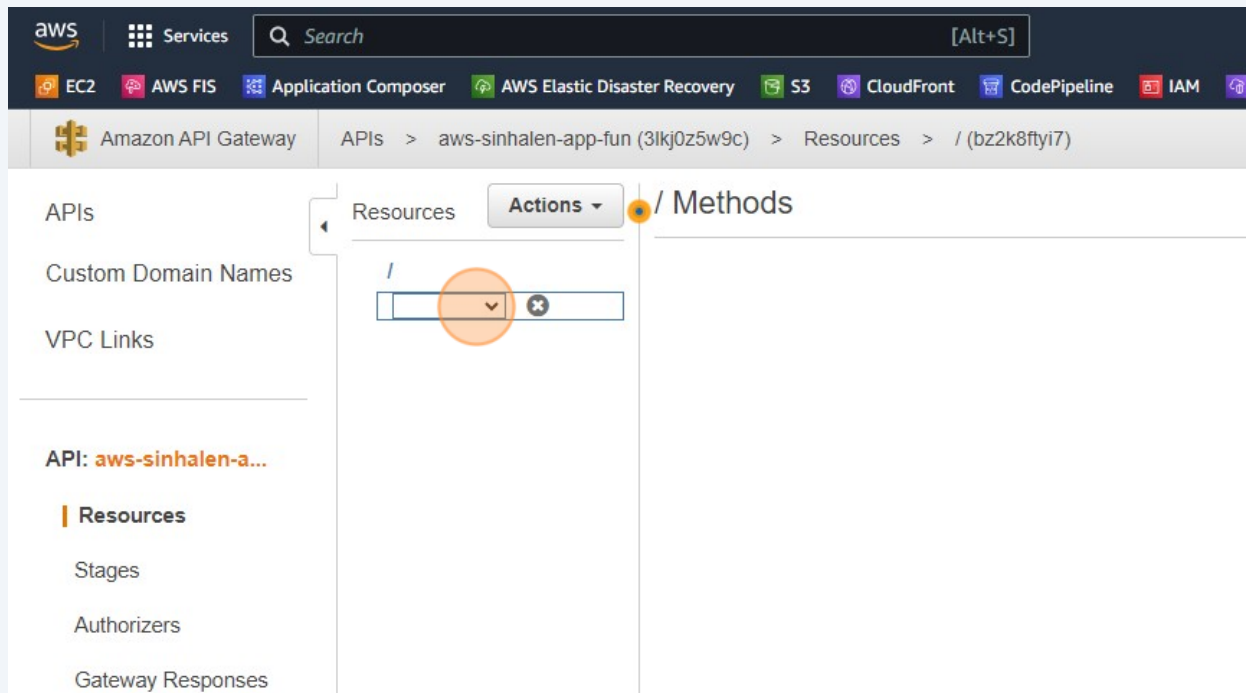


## 16 Click "Create Method"

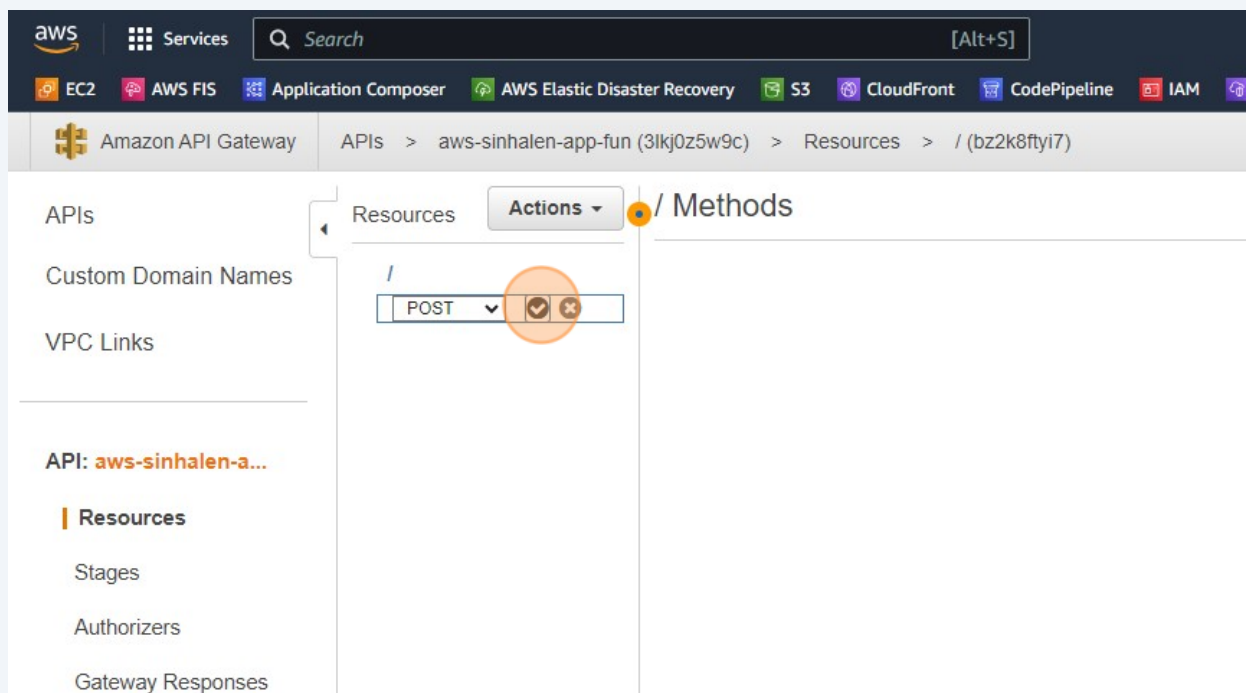




## 17 Click this dropdown.



## 18 Click "Confirm creating Method"



19 Click "aws-sinhalen-app-f"

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

Use Lambda Proxy integration ☐ ⓘ

Lambda Region

Lambda Function

Use Default Timeout

20 Click "Save"

ⓘ

21

22



## 23 Click "Deploy API"

The screenshot shows the AWS API Gateway console interface. On the left, a sidebar contains navigation links: Custom Domain Names, VPC Links, API: aws-sinhalen-a..., Resources (highlighted), Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, Dashboard, and Settings. The main area displays a 'POST' method. A context menu is open over the 'POST' method, showing 'RESOURCE ACTIONS' (Edit Method Documentation, Delete Method) and 'API ACTIONS' (Create Method, Create Resource, Enable CORS, Edit Resource Documentation, **Deploy API** (highlighted with an orange circle), Import API, Edit API Documentation, Delete API). To the right, the 'Method Request' section shows 'Auth: NONE' and 'ARN: arn:aws:execute-api:us-east-1:102791069024:3lkj0z5w9c/\*/\*/\*'. Below it, the 'Method Response' section shows 'HTTP Status: 200' and 'Models: application/json => Empty'.

## 24 Click "Deploy"

The screenshot shows the AWS API Gateway console with a deployment modal open. The modal contains the following fields: 'Deployment stage' (dropdown menu with '[New Stage]' selected), 'Stage name\*' (text input with 'dev'), 'Stage description' (text input), and 'Deployment description' (text input). At the bottom of the modal are 'Cancel' and 'Deploy' buttons, with the 'Deploy' button highlighted by an orange circle. The background shows a partial view of the API Gateway console, including the 'Integration Response' section with 'HTTP status pattern:' and 'Output passthrough: No'.

**25** Thank You !!