

## Overview

API or Application Programming Interface is like a messenger that lets two different programs to talk to each other. For example, assume you are at a restaurant, you sit at a table and order food from a waiter, who takes your requests to the cook and brings back your meal.

So, in the above case waiter is like an API which communicates with the system without needing to know how the kitchen works.

## Project Details

1. Create an API Gateway
2. Build API Gateway with Lambda Integration

## Section 1: Create an API Gateway

1. Login into AWS Console
2. Click on Create API
3. Click on REST API
4. Click on Build
5. Ignore the pop-up
6. Select New API
7. Enter API Name
8. Click on Create API
9. Click the APIs under API Gateway and the API is created
10. Screenshots – Refer Appendix

## Section 2: Build API Gateway with Lambda Integration

1. Login into AWS Console
2. Navigate to AWS Lambda and click on Create a Function
3. Select Author from scratch
4. Provide a function name
5. Select runtime as python
6. Click on Create a New Role from Aws policy templates
7. Provide a role name
8. Select basic lambda@edge permissions – Minimum Permissions
9. The role allows or grants permissions to execute and access resources it might need, like reading from S3 or writing to logs
10. Click on Create Function

11. Once the function is created, follow the steps as in section 1 – Create an API
12. Click on Create an API
13. Click on REST API – Build
14. Click on New API
15. Provide a API Name and Click on Create API
16. Now create a resource – Refer Screen shots
17. Click on Resources and Click on Create Resource
18. Provide the resource name – myfirstapi
19. Click on create resource
20. After the resource has been created click on Create Method
21. Select Method Type as GET
22. Select Integration Type as Lambda Function
23. Under LambdaFunction – Select the Region and the ARN or Select from the Drop Down
24. Click on Create Method
25. Click on Deploy the API
26. Select the Stage Name as New Stage
27. Provide a Stage Name
28. Click on Deploy
29. Copy the Invoke URL as shown and paste it in the browser
30. At the end add / and type myfirstapi and enter
31. You will get a Get request

# Appendix

## Section 1: Screen Shots

### Get started

Create a new API to begin exploring API Gateway. You can also import an external definition file into API Gateway.

Create an API

### REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:  
Lambda, HTTP, AWS Services

ImportBuild

### 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

myfirstapi

Description - optional

myfirstapi

API endpoint type

Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

Regional

IP address type

Select the type of IP addresses that can invoke the default endpoint for your API.

☒ IPv4  
Supports only edge-optimized and Regional API endpoint types.

☐ Dualstack  
Supports all API endpoint types.

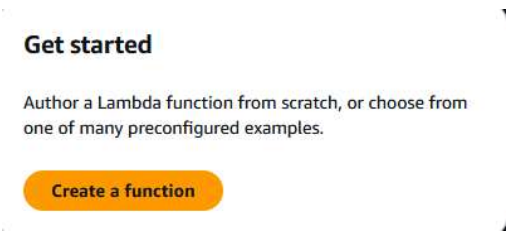
CancelCreate API

### APIs (1/1)

Find APIs

Name	Description	ID	Protocol	API endpoint type	Created
myfirstapi	myfirstapi	68a6xqno0d	REST	Regional	2025-04-23

## Section 2: Screen Shots



**Create function** [info](#)

Choose one of the following options to create your function.

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

☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**  
Select a container image to deploy for your function.

---

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
  
Function names must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

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

**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 triggers.

---

**▼ Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

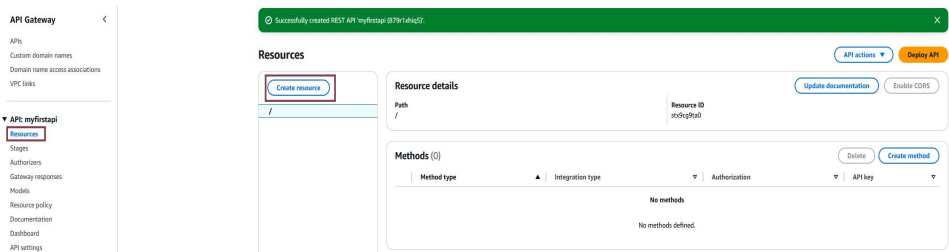
☐ Create a new role with basic Lambda permissions  
☐ Use an existing role  
☒ **Create a new role from AWS policy templates**

[🔄](#) Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

**Role name**  
Enter a name for your new role.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Policy templates - optional** [info](#)  
Choose one or more policy templates.  
 [🔄](#)  
CloudWatch Logs

## Section 2A: API Gateway Creation



Create resource

Resource details

Proxy resource

Info

Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path

/

Resource name

myfirstapi

☐ CORS (Cross Origin Resource Sharing)

Info

Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel

Create resource

Methods (0)

Delete

Create method

Method type

Integration type

Authorization

API key

No methods

No methods defined.

Resources

Create resource

/

/myfirstapi

GET

API actions

Deploy API

/myfirstapi - GET - Method execution

Update documentation

Delete

ARN

arn:aws:apigateway:us-east-1::/restapis/myfirstapi

Resource ID

odj5g1

Client

Method request

Integration request

Method response

Integration response

Lambda

Integration

Deploy API

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)

Stage

\*New stage\*

Stage name

myfirstapideploy

A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description

myfirstapideploy

Cancel

Deploy

Stages

myfirstapideploy

/myfirstapi

GET

Stage details

Stage name

myfirstapideploy

Rate

10000

Web ACL

-

Cache cluster

inactive

Burst

5000

Client certificate

-

Default method-level caching

inactive

Invoke URL

myfirstapideploy

Active deployment

rbkors on April 23, 2025, 20:45 (UTC+05:30)

Logs and tracing

CloudWatch logs

inactive

Detailed metrics

inactive

X-Ray tracing

inactive

Data tracing

inactive

Custom access logging

inactive

Stage variables

Deployment history

Documentation history

Canary

Tags

Stage variables (0/0)

Find resources

Name

Value

No variables

No variables associated with the stage.

Manage variables

https://879r1xhiq5.execute-api.us-east-1.amazonaws.com/myfirstapideploy/myfirstapi

← → ↺

🔒

https://879r1xhiq5.execute-api.us-east-1.amazonaws.com/myfirstapideploy/myfirstapi

JSON

Raw Data

Headers

Save

Copy

Pretty Print

```
{"statusCode": 200, "body": "\"Hello from Lambda!\""} 
```