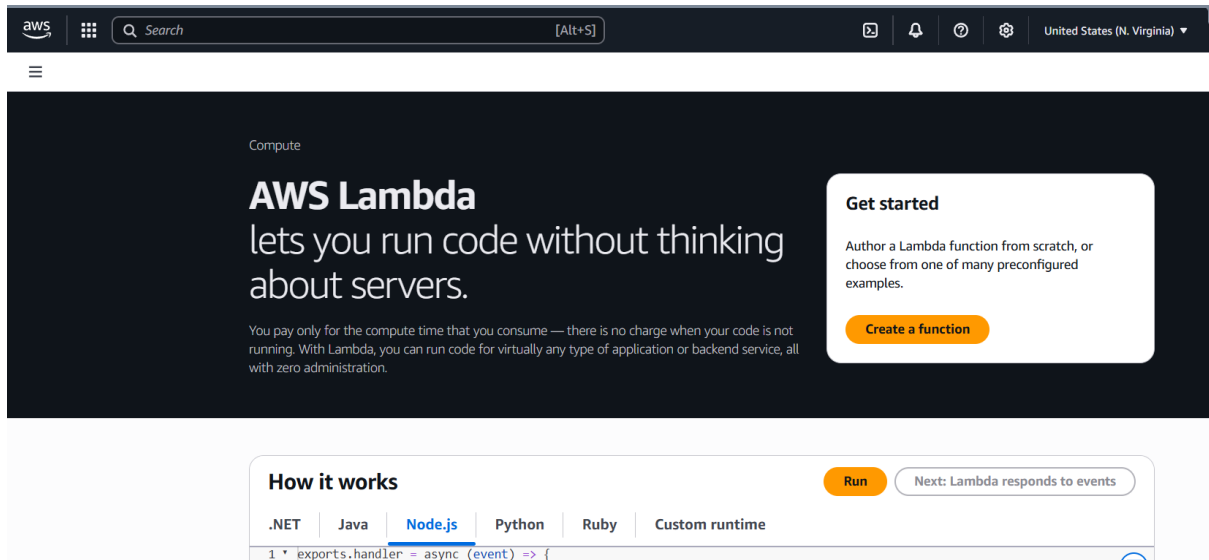


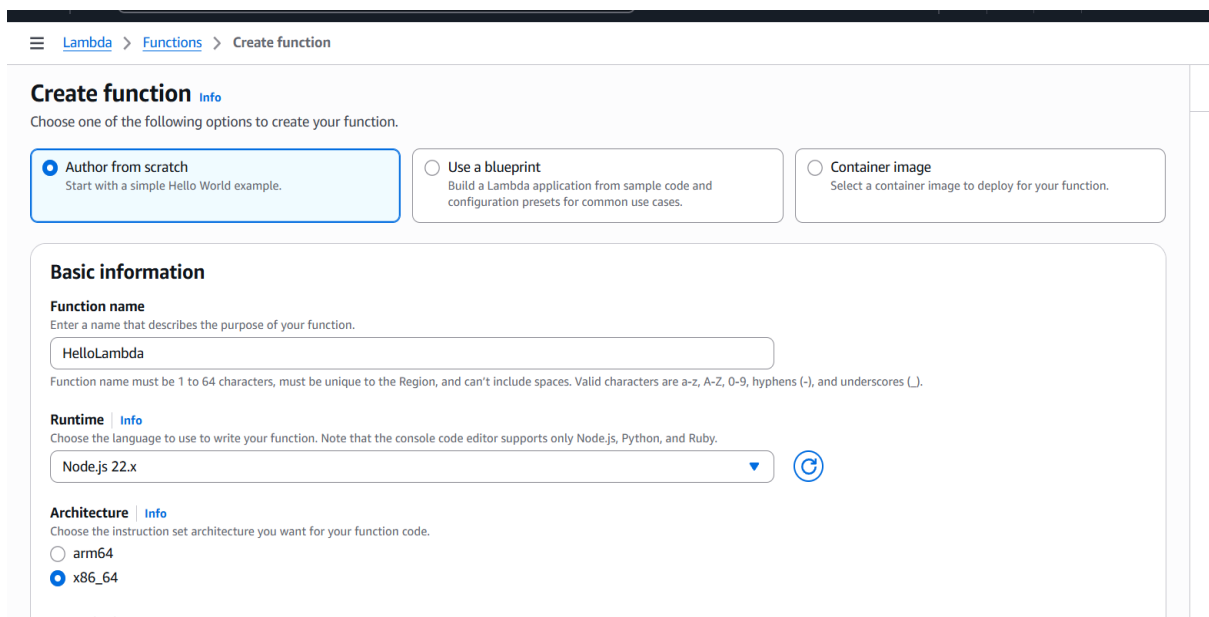
API GATEWAY:

Go to lambda function and create a lambda function.

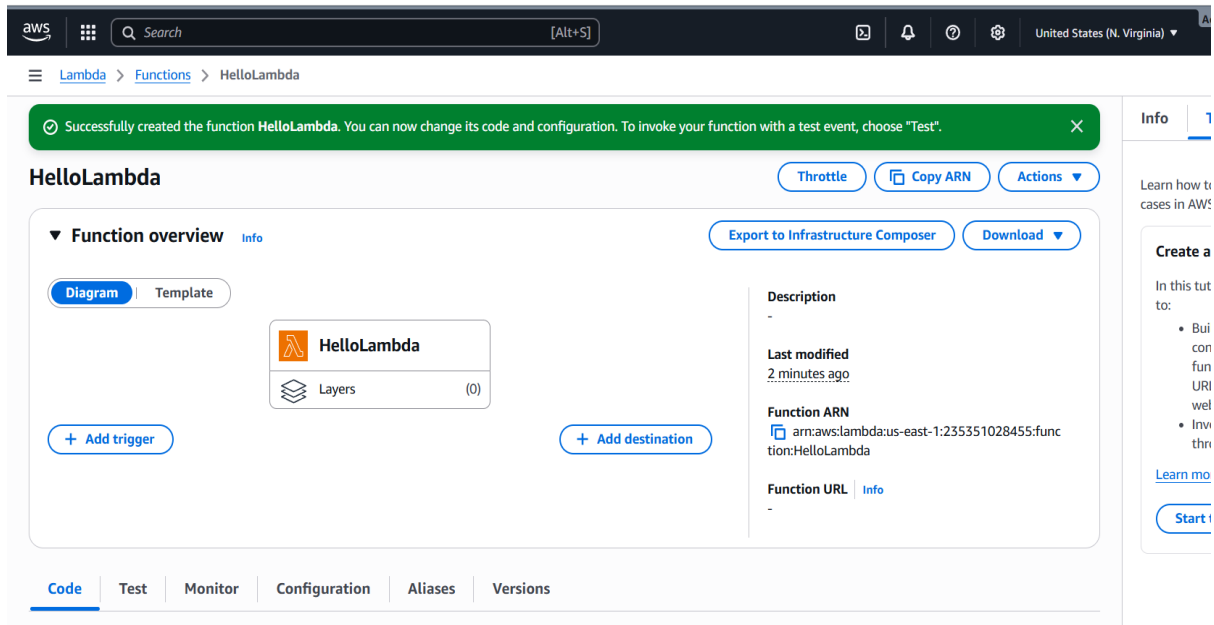


The image shows the AWS Lambda landing page. At the top is the AWS navigation bar with the logo, a search bar, and various icons. Below the navigation bar is a dark banner with the text "Compute" and "AWS Lambda lets you run code without thinking about servers." To the right of the banner is a "Get started" box with a "Create a function" button. Below the banner is a "How it works" section with tabs for .NET, Java, Node.js, Python, Ruby, and Custom runtime. The Node.js tab is selected, and a code editor shows a snippet of JavaScript code: `exports.handler = async (event) => {`.

Give function as author from scratch, give name and mention run time and save.

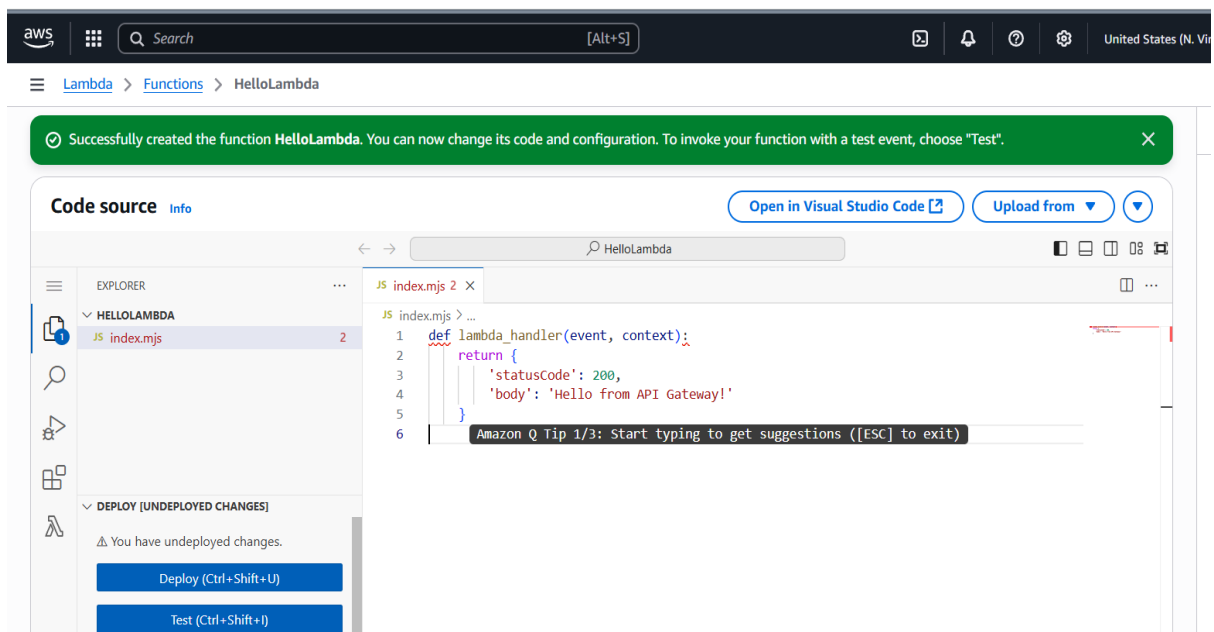


The image shows the "Create function" form in the AWS Lambda console. The breadcrumb navigation is "Lambda > Functions > Create function". The form has three tabs: "Function name", "Runtime", and "Architecture". The "Function name" tab is active, showing a text input field with "HelloLambda" and a description: "Enter a name that describes the purpose of your function." Below the input field is a note: "Function name 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 (_)." The "Runtime" tab is also visible, showing a dropdown menu with "Node.js 22.x" selected and a note: "Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby." The "Architecture" tab is also visible, showing two radio buttons: "arm64" and "x86_64", with "x86_64" selected. A "Create function" button is at the bottom right.



```
def lambda_handler(event, context):  
    return {  
        'statusCode': 200,  
        'body': 'Hello from API Gateway!'  
    }
```

Write this code and deploy it.



Go to api gateway

The screenshot shows the AWS API Gateway console. At the top, there's a navigation bar with the AWS logo, a search bar, and account information (United States (N. Virginia), Account ID: 2353-5102-8455). The main header area has the text "Networking & Content Delivery" and "API Gateway" in large font, followed by "Create and manage APIs at scale". Below this, a paragraph describes Amazon API Gateway as a fully managed service. To the right, a "Get started" section encourages creating a new API. A central "How it works" diagram illustrates the flow from client applications (web, mobile, IoT) through the API Gateway to various backends (AWS Lambda, Amazon S3, Amazon Kinesis, Amazon DynamoDB, Amazon ElastiCache, Amazon SNS, Amazon SQS, Amazon CloudWatch, and other AWS services). A "Pricing" section on the right explains the pay-per-use model for different API types.

Click on create API

Select rest api click on build.

This screenshot shows the "Create API" page in the AWS API Gateway console. The breadcrumb navigation at the top reads "API Gateway > APIs > Create API". The page lists three API types: "WebSocket API", "REST API", and "REST API Private". The "REST API" section is highlighted with a red arrow pointing to its "Build" button. Each section includes a brief description and a list of services it works with (Lambda, HTTP, AWS Services).

Click on create new api, give the api name, select regional click on create.

aws

Search

[Alt+S]

API Gateway > APIs > Create API > Create REST API

☒ New API
Create a new REST API.

☐ Import API
Import an API from an OpenAPI definition.

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

☐ Example API
Learn about API Gateway with an example.

API name
HelloAPI

Description - optional

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 VPC.
Regional

IP address type [Info](#)
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.

Click on resource and click create resource.

aws

Search

[Alt+S]

United States (N. Virginia)

Account ID: 2353-5102-

API Gateway > APIs > Resources - HelloAPI (bvtktxugib)

API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
▼ API: HelloAPI
Resources
Stages
Authorizers
Gateway responses
Models
Resource policy
Documentation
Dashboard
API settings

Successfully created REST API 'HelloAPI (bvtktxugib)'.

Resources

Create resource

/

Resource details
Path
/
Resource ID
gdiqfubhil

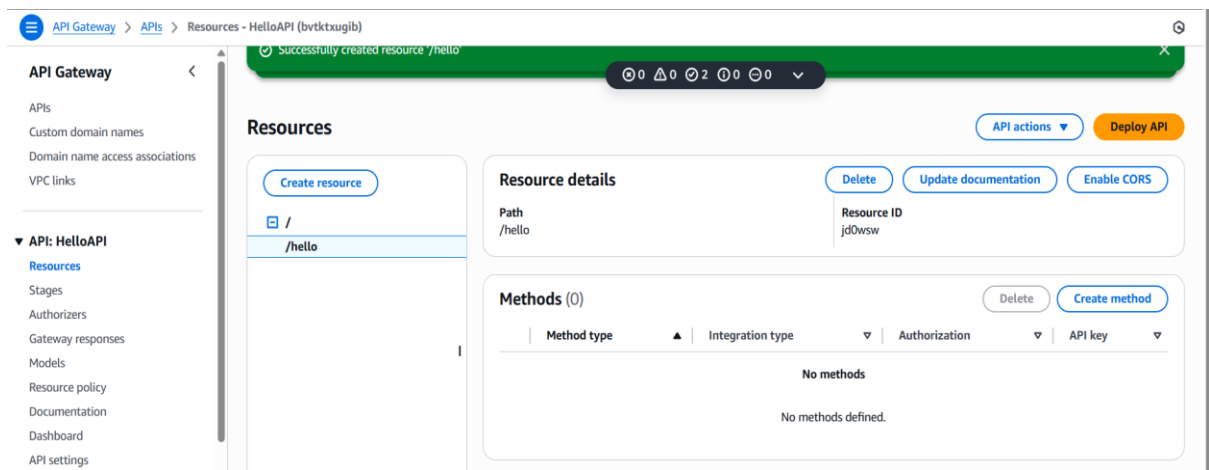
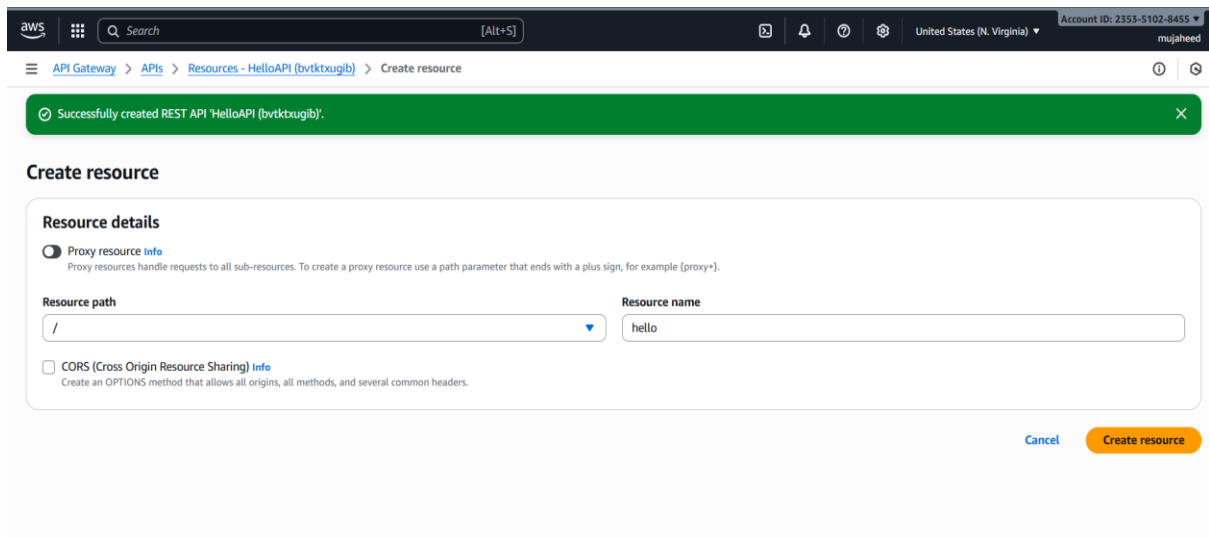
API actions

Deploy

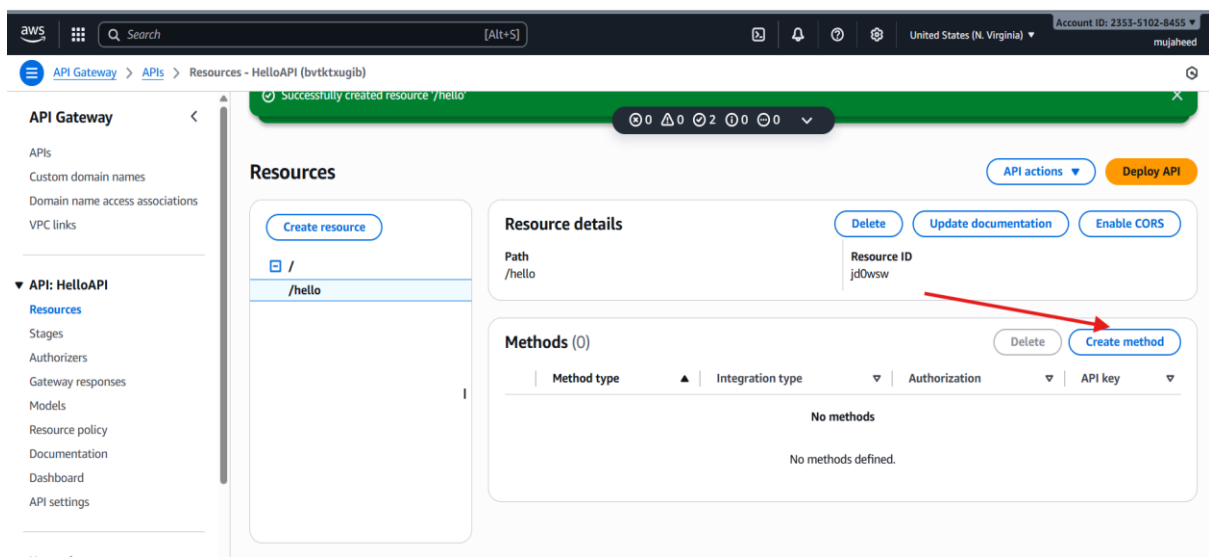
Update documentation

Enable CORS

Methods (0)
Delete Create method
Method type Integration type Authorization API key
No methods
No methods defined.



Click on create method



- Give the method as GET
- Select lambda function
- Select your lambda function.

Method details

Method type
GET

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.
- ☐ **AWS service**
Integrate with an AWS Service.
- ☐ **VPC link**
Integrate with a resource that isn't accessible over the public internet.

☒ **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.

us-east-1

API Gateway

APIs
Custom domain names
Domain name access associations
VPC links

▼ **API: HelloAPI**

- Resources
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

Resources

Create resource

- /
- /hello
- GET

/hello - GET - Method execution

ARN: arn:aws:execute-api:us-east-1:235351028455:bvktkxugib/*
GET/hello

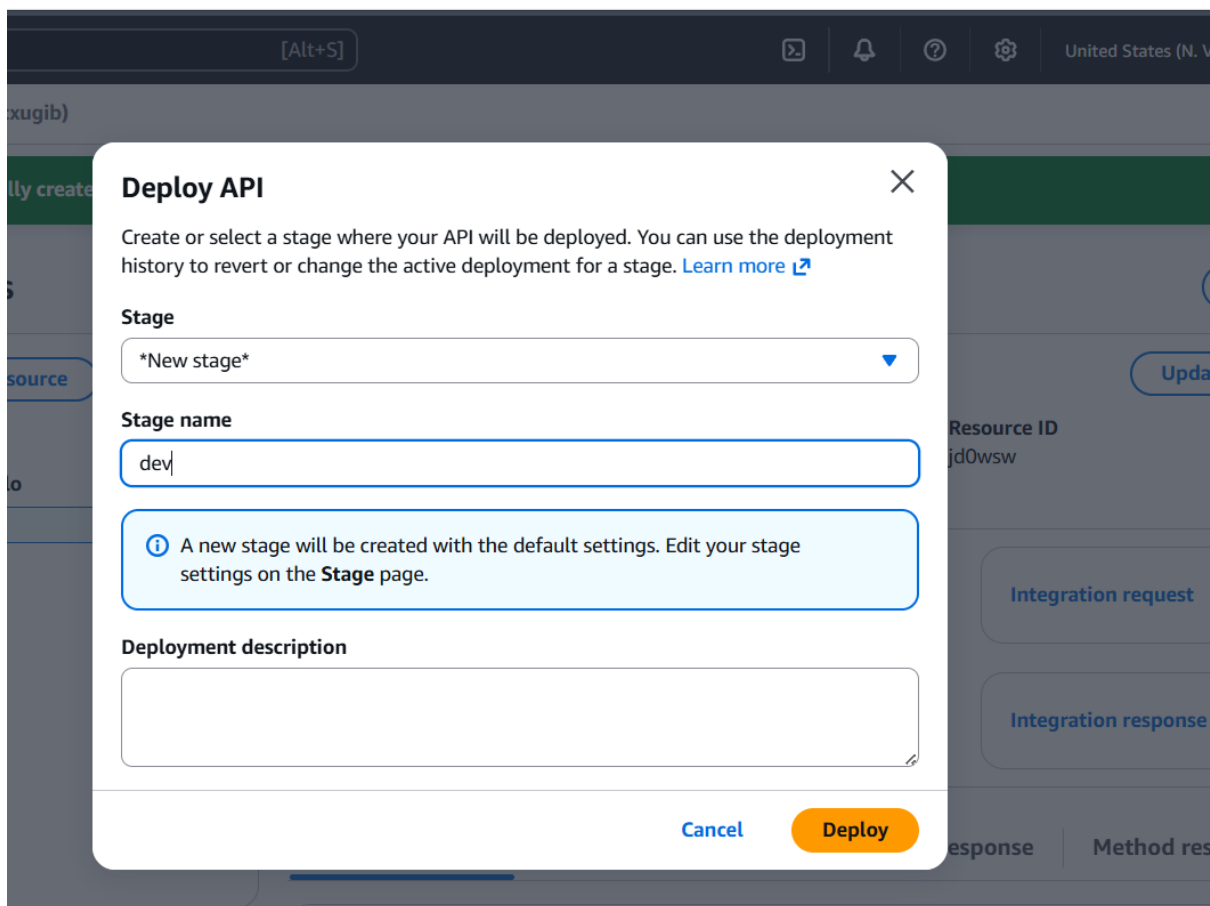
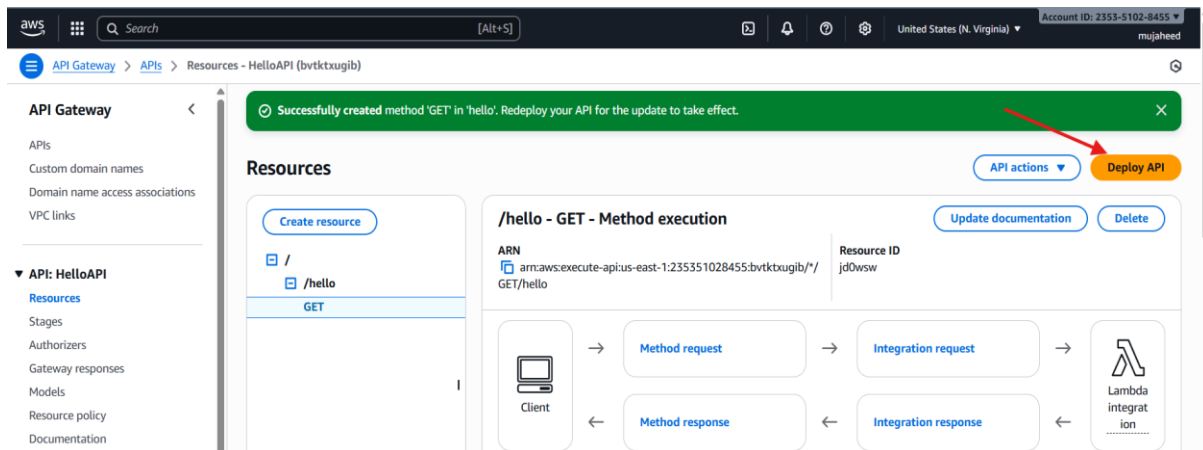
Resource ID: jd0wsw

Update documentation Delete

Client → Method request → Integration request → Lambda integration → Integration response → Method response → Client

Method request Integration request Integration response Method response Test

Click on deploy api



It will give invoke URL copy that URL and paste on browser and add /hello at last it will give you Hello from API "Gateway!"

aws

Search

[Alt+S]

United States (N. Virginia)

Account ID: 2353-5102-8455

mujahed

API Gateway

APIs

HelloAPI (bvtktxugib)

Stages

API Gateway

APIs

Custom domain names

Domain name access associations

VPC links

API: HelloAPI

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Stages

dev

Stage details

Stage name

dev

Cache cluster

Inactive

Default method-level caching

Inactive

Rate

10000

Burst

5000

Web ACL

-

Client certificate

-

Invoke URL

https://bvtktxugib.execute-api-us-east-1.amazonaws.com/dev

Active deployment

73tgi3 on October 29, 2025, 15:46 (UTC+05:30)

Loos and tracing