Serverless

AWS Lambda

Serverless

A cloud-computing execution model in which the cloud provider runs the server

The provider dynamically manages the allocation of machine resources.

Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.

AWS Lambda

A service which performs serverless computing

Benefits

NO SERVER TO MANAGE

BILLING BASED ON USAGE

CONTINUOUS SCALING

Pay per Request

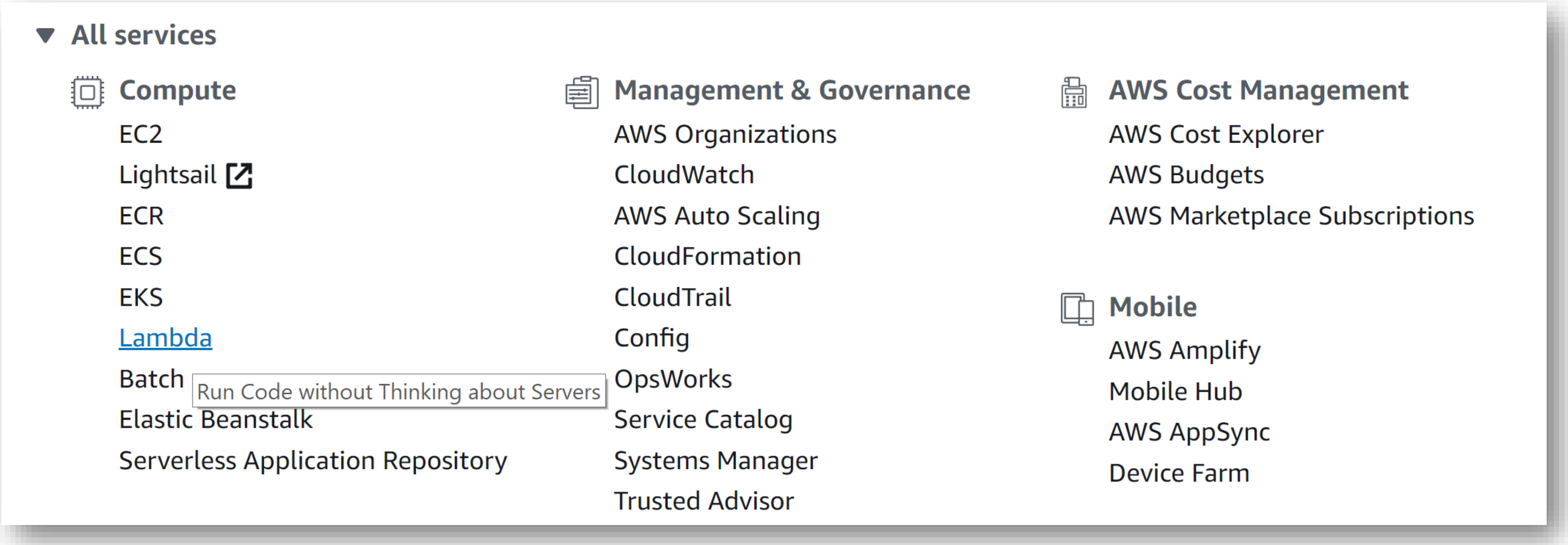
- Low request cost
- No hourly, daily, or monthly minimum charge
- No per device fees
- Never pay for idle

Activity-1

Register on AWS

https://aws.amazon.com/lambda/

Let’s Create a Lambda Function



▼ All services

Compute

EC2

Lightsail

ECR

ECS

EKS

Lambda

Run Code without Thinking about Servers

Batch

Elastic Beanstalk

Serverless Application Repository

Management & Governance

AWS Organizations

CloudWatch

AWS Auto Scaling

CloudFormation

CloudTrail

Config

OpsWorks

Service Catalog

Systems Manager

Trusted Advisor

AWS Cost Management

AWS Cost Explorer

AWS Budgets

AWS Marketplace Subscriptions

Mobile

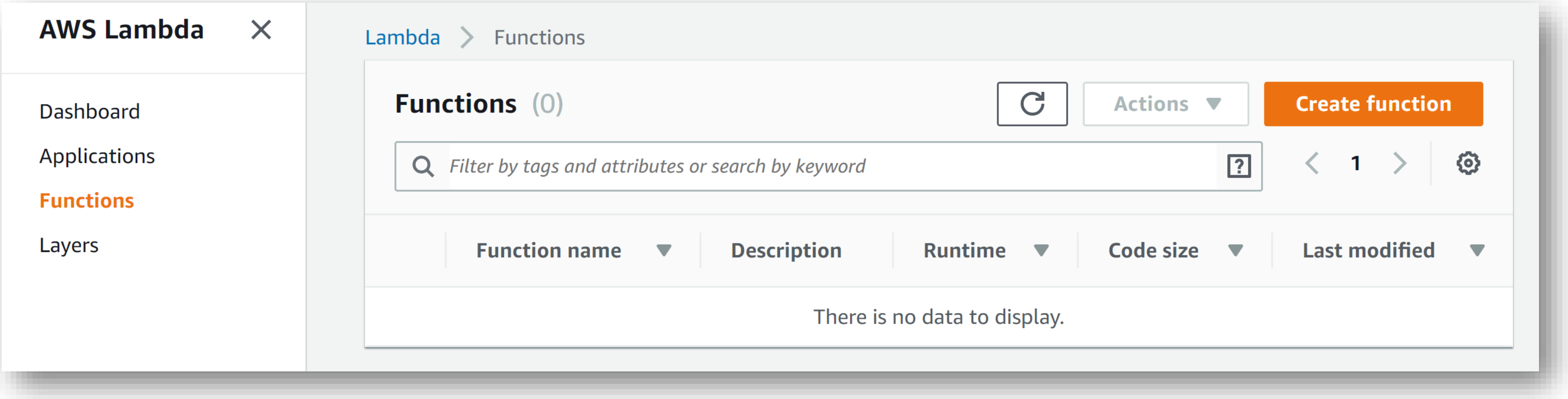
AWS Amplify

Mobile Hub

AWS AppSync

Device Farm

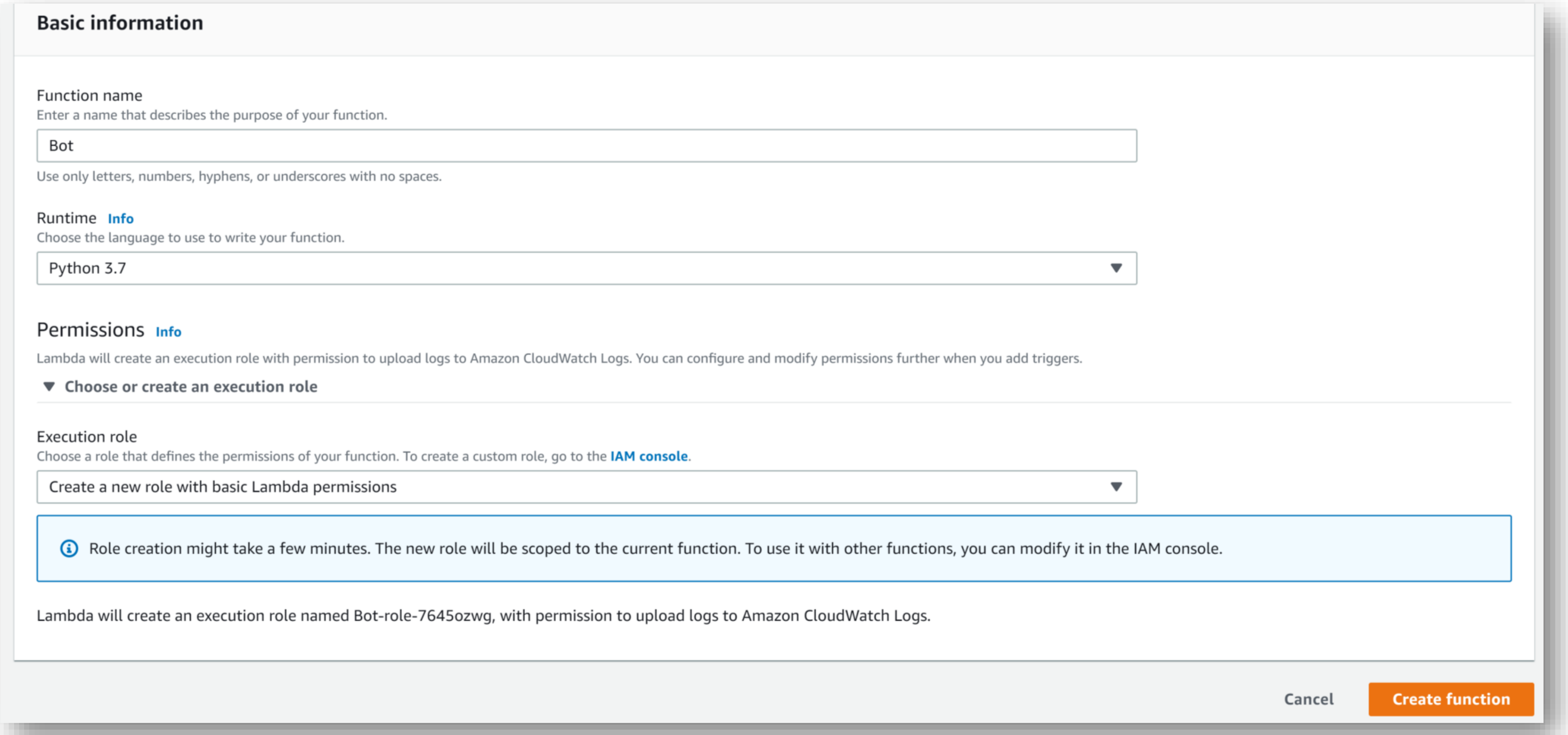
Let’s Create a Lambda Function



Function name	Description	Runtime	Code size	Last modified

There is no data to display.

Let’s Create a Lambda Function



Basic information

Function name

Enter a name that describes the purpose of your function.

Bot

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

Runtime

Choose the language to use to write your function.

Python 3.7 ▼

Permissions

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an execution role

Execution role

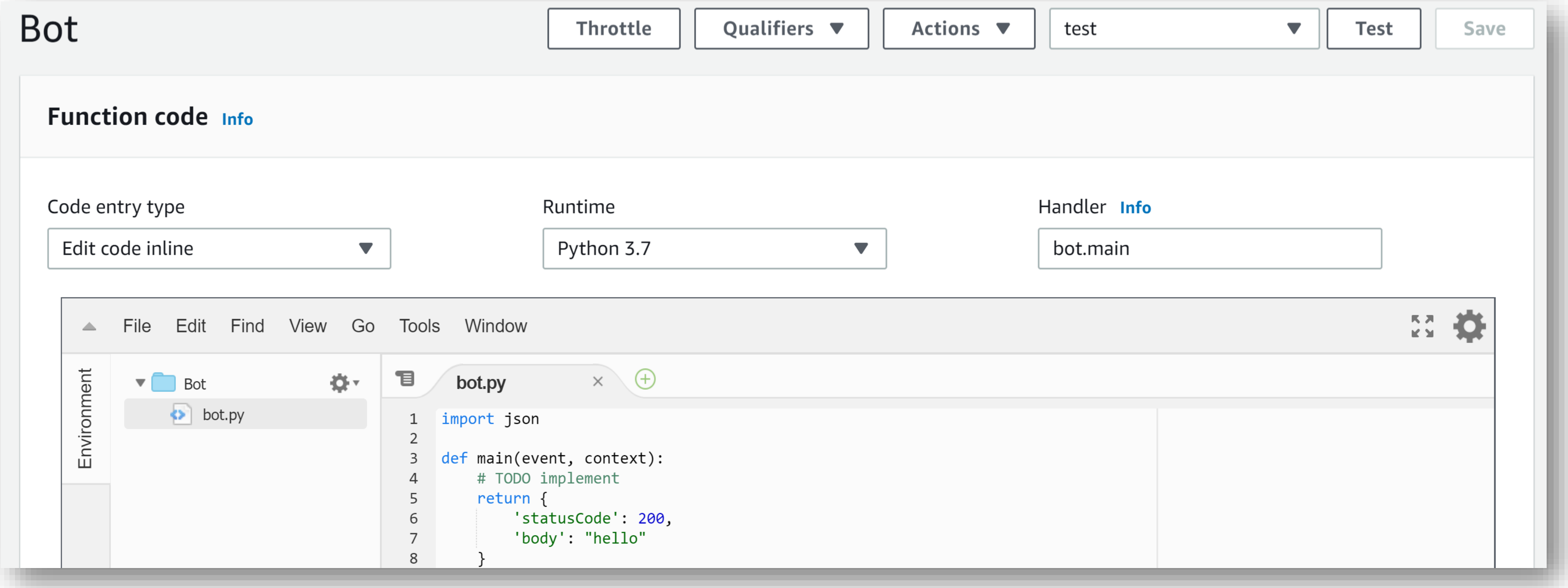
Choose a role that defines the permissions of your function. To create a custom role, go to the **IAM console**.

Create a new role with basic Lambda permissions ▼

ⓘ Role creation might take a few minutes. The new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Lambda will create an execution role named Bot-role-7645ozwg, with permission to upload logs to Amazon CloudWatch Logs.

Let’s Create a Lambda Function



Code entry type: Edit code inline

Runtime: Python 3.7

Handler: bot.main

```
import json

def main(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': "hello"
    }
```

Function Handler

- **Event:** represents the trigger that caused the invocation of the lambda. For example, if your lambda is triggered by an API call, it contains the payload of request.

- **Context:** argument provides methods and properties that provide information about the invocation, function, and execution environment (memory, remaining time, etc.)

Activity-2

Write a simple bot

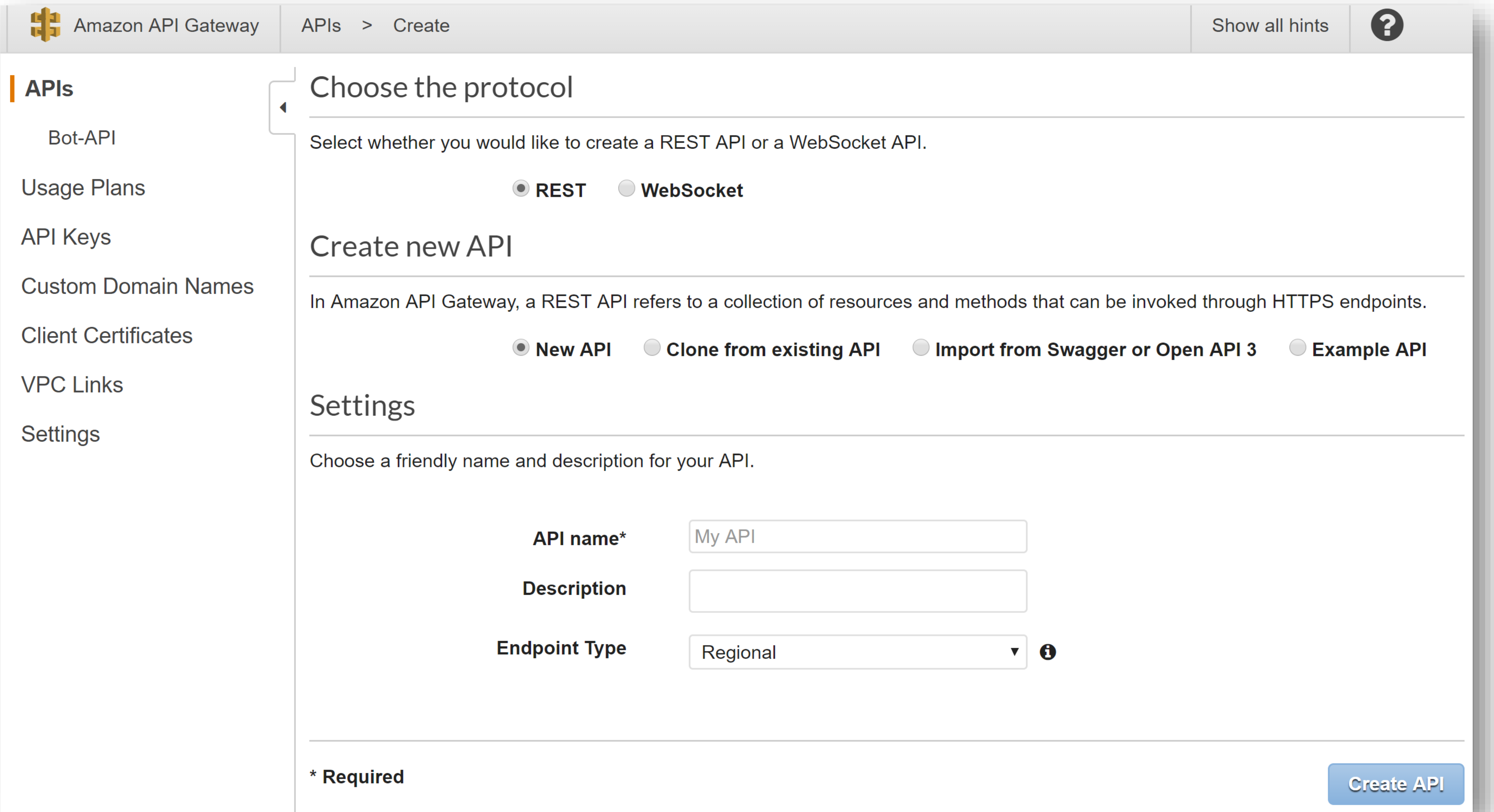
- if the payload contains “hello” it returns “hello”
- if it contains “how are you?” it returns “fine!”

Test the code

Connect to the world

- https://us-east-2.console.aws.amazon.com/apigateway/
- Create a new API
- Design Resources
- Deploy API

Create a 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 ○ Clone from existing API ○ Import from Swagger or Open API 3 ○ Example API

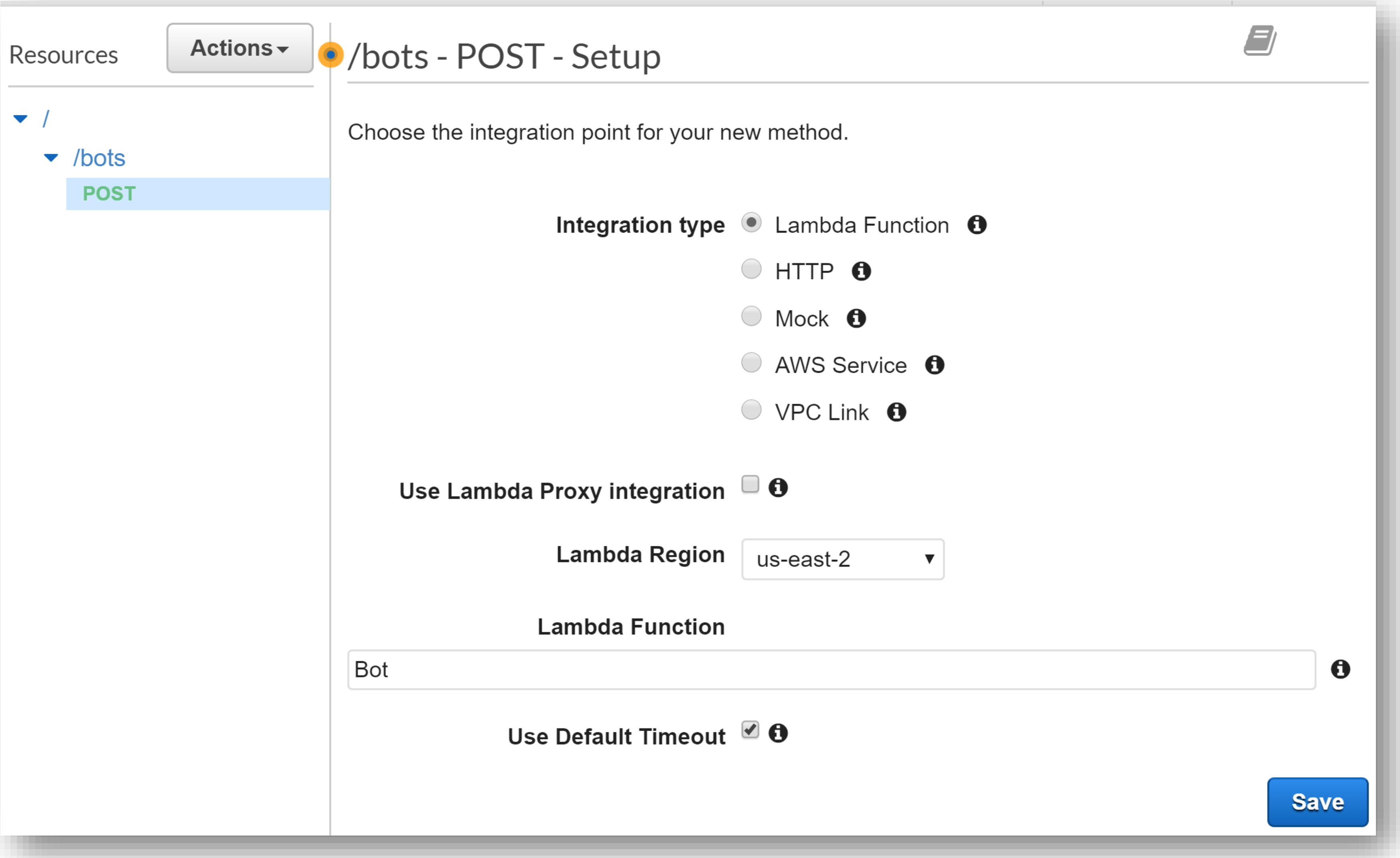
Settings

Choose a friendly name and description for your API.

API name*	My API
Description	
Endpoint Type	Regional ▾

* Required

Create Resources



/bots - POST - Setup

Choose the integration point for your new method.

Integration type ● Lambda Function
○ HTTP
○ Mock
○ AWS Service
○ VPC Link

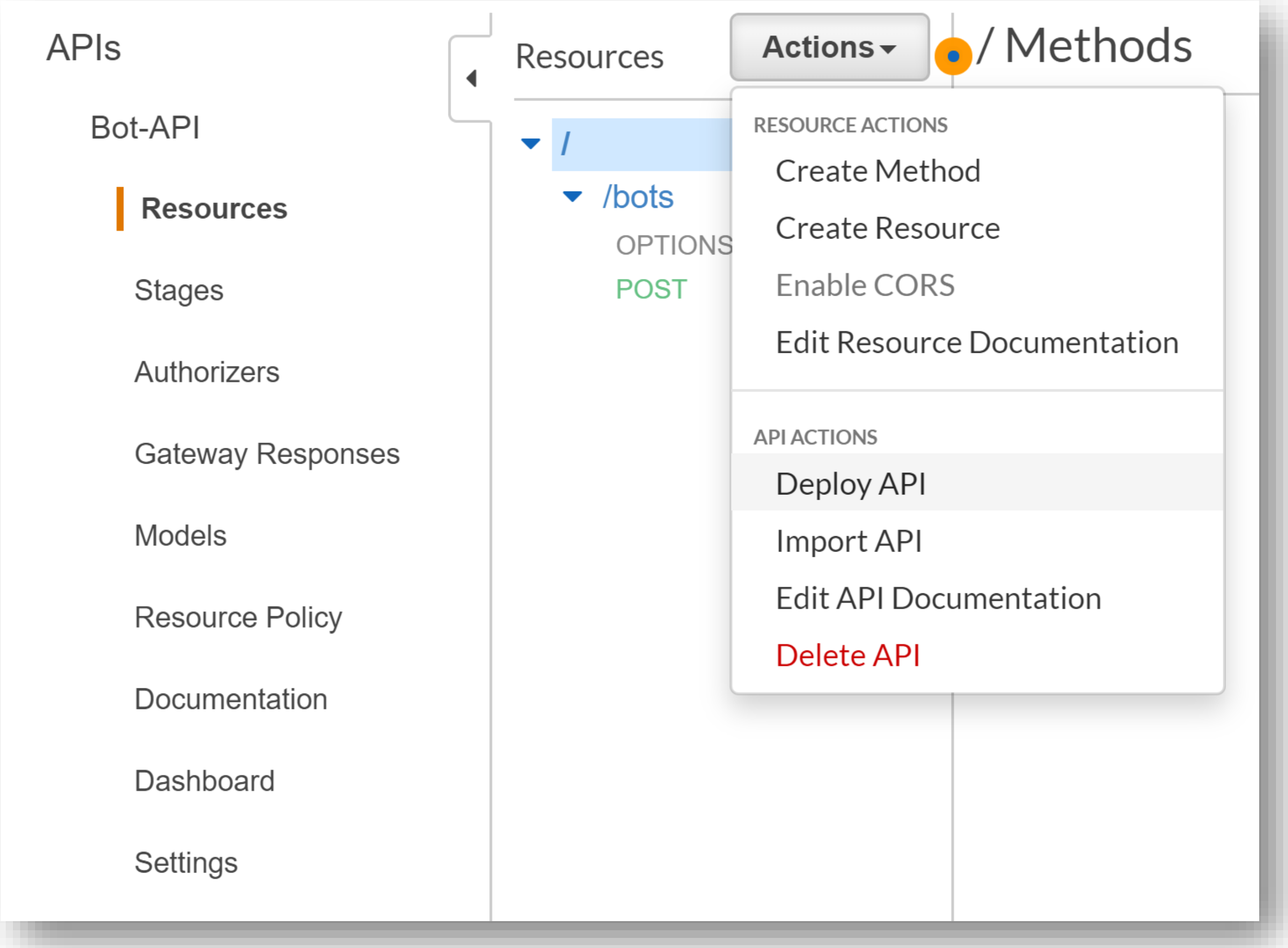
Use Lambda Proxy integration □

Lambda Region us-east-2

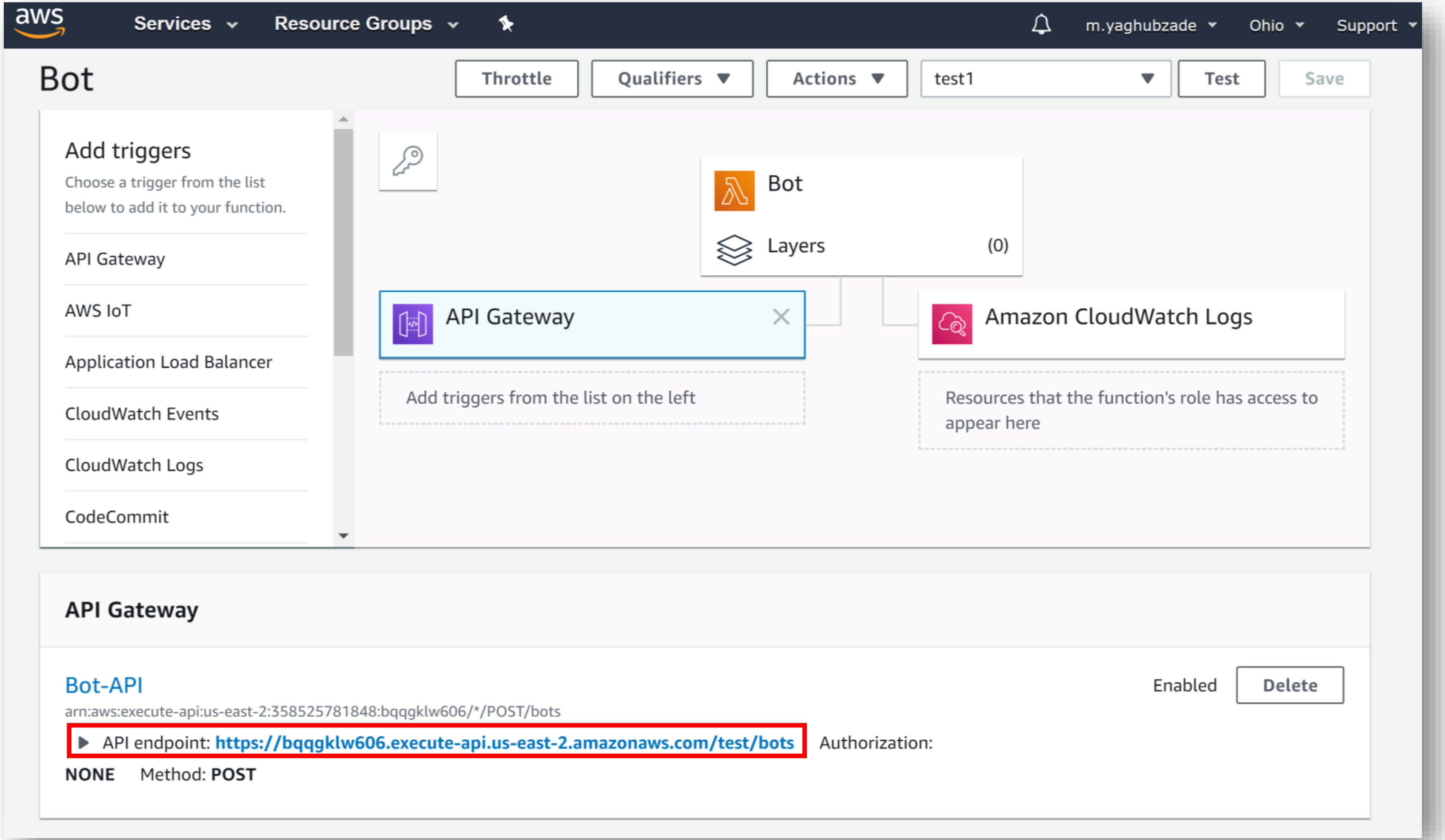
Lambda Function Bot

Use Default Timeout ☑

Deploy API


APIs
Bot-API
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Resources
Actions
/ Methods
/
/bots
OPTIONS
POST
RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Edit Resource Documentation
API ACTIONS
Deploy API
Import API
Edit API Documentation
Delete API

Test Your API



Bot

Add triggers

Choose a trigger from the list below to add it to your function.

- API Gateway
- AWS IoT
- Application Load Balancer
- CloudWatch Events
- CloudWatch Logs
- CodeCommit

Add triggers from the list on the left

Resources that the function's role has access to appear here

API Gateway

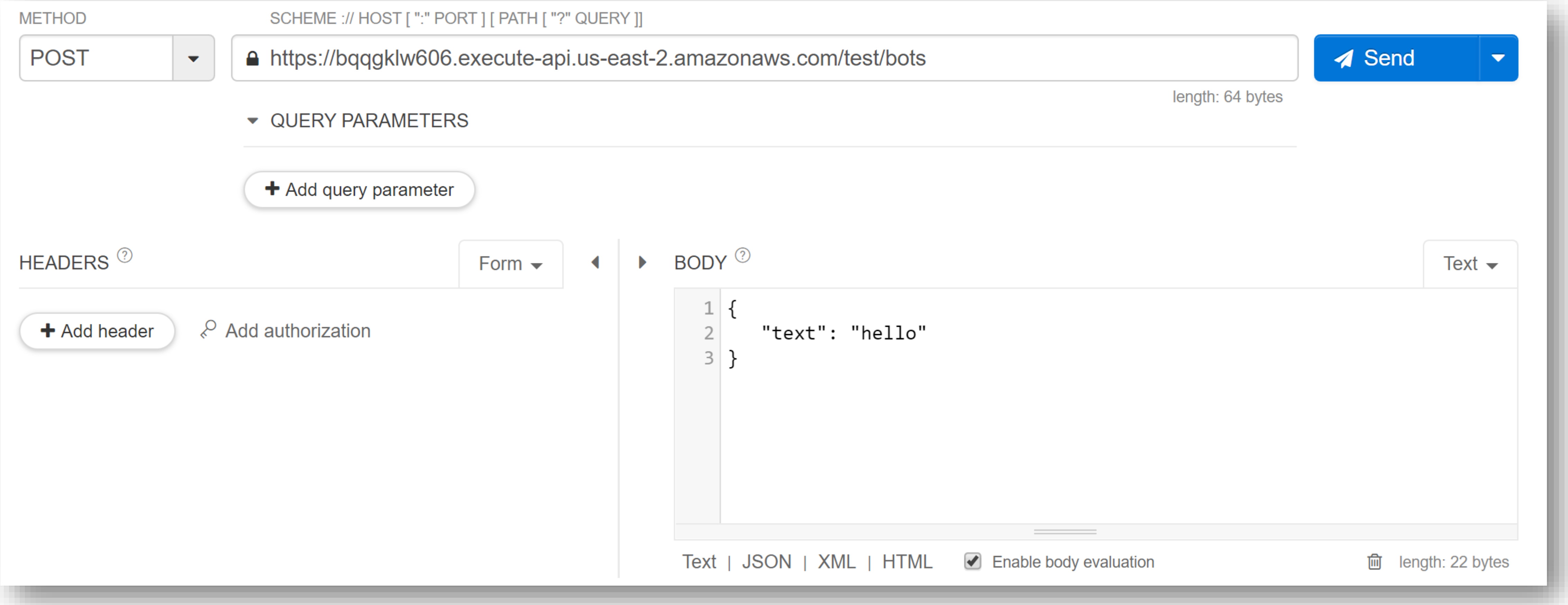
Bot-API

arn:aws:execute-api:us-east-2:358525781848:bqqgklw606/*/POST/bots

Enabled Delete

▶ API endpoint: **https://bqqgklw606.execute-api.us-east-2.amazonaws.com/test/bots** Authorization: **NONE** Method: **POST**

Test Your API



METHOD

POST

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]

https://bqqgklw606.execute-api.us-east-2.amazonaws.com/test/bots

length: 64 bytes

```
{
    "text": "hello"
}
```

☑ Enable body evaluation

length: 22 bytes