

Creating a Serverless API

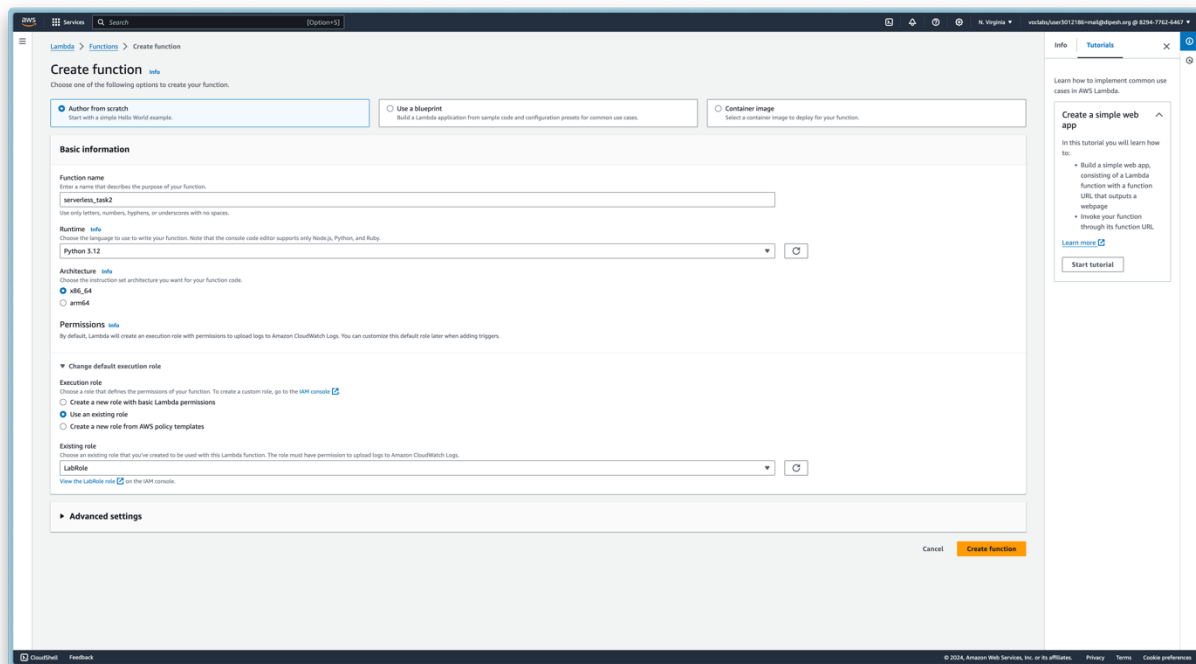
Objective: Develop a serverless API using AWS Lambda and API Gateway.

Approach:

- **Define API:** Design a simple RESTful API (e.g., for a todo list application).
- **Lambda Functions:** Create Lambda functions for each API method (GET, POST, PUT, DELETE).
- **API Gateway Setup:** Use API Gateway to set up the API endpoints, connecting each endpoint to the corresponding Lambda function.
- **Testing:** Test the API using tools like Postman or AWS API Gateway test functionality.

Goal: Gain hands-on experience in building and deploying a serverless API, understanding the integration between Lambda and API Gateway.

Create a Lambda function



The screenshot shows the AWS Lambda 'Create function' console page. The page is titled 'Create function' and has a sub-header 'Choose one of the following options to create your function.' There are three tabs: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Author from scratch' tab is active, showing the 'Basic information' section. In this section, the 'Function name' is 'serverless_todo2', the 'Runtime' is 'Python 3.12', and the 'Architecture' is 'x86_64'. The 'Permissions' section is expanded, showing 'Change default execution role' with options to 'Create a new role with basic Lambda permissions' (selected), 'Use an existing role', or 'Create a new role from AWS policy templates'. The 'Existing role' dropdown is set to 'LabRole'. At the bottom right, there are 'Cancel' and 'Create function' buttons. A sidebar on the right contains a 'Tutorials' section with a link to 'Create a simple web app'.

Create a dynamodb table

Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Share feedback

DynamoDB > Tables > Create table

Create table

Table details [info](#)
DynamoDB is a NoSQL database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
serverless_task2
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across nodes for availability and availability.
id String 1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name String 1 to 255 characters and case sensitive.

Table settings

☒ **Default settings**
This fastest way to create your table. You can modify these settings now or after your table has been created.

☐ **Customize settings**
Use these advanced features to make DynamoDB work better for your needs.

Default table settings
These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value	Editable after creation
Table class	DynamoDB Standard	Yes
Capacity mode	Provisioned	Yes
Provisioned read capacity	5 RCU	Yes
Provisioned write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes

Create a Rest API

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.

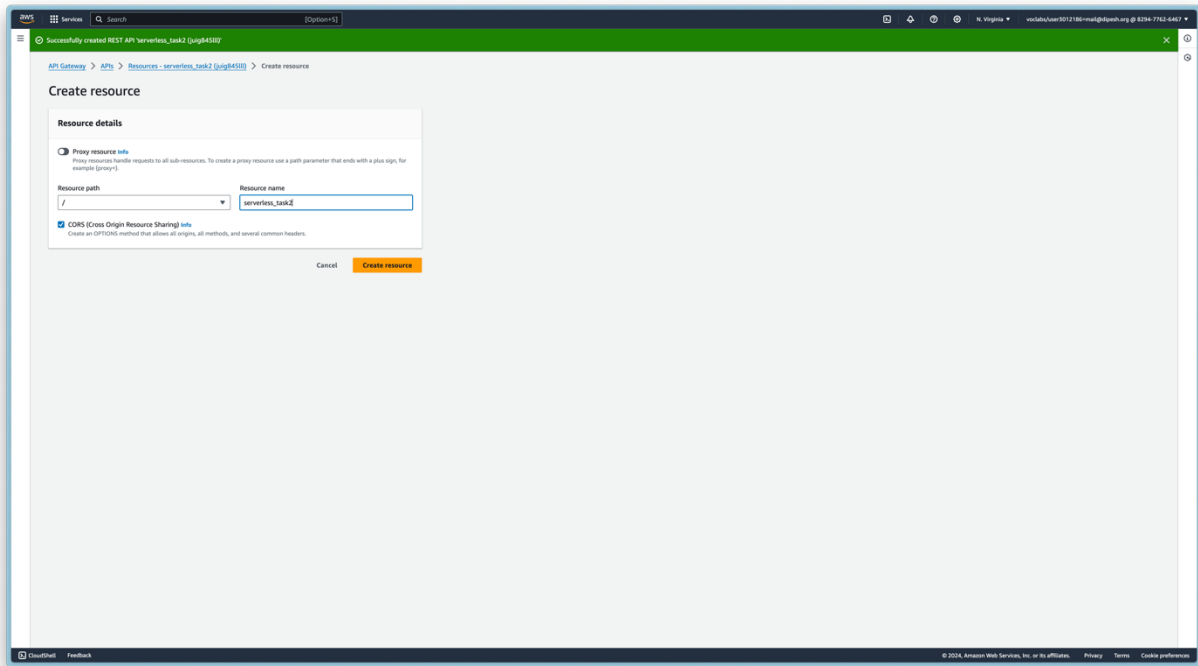
API name
serverless_task2

Description - optional

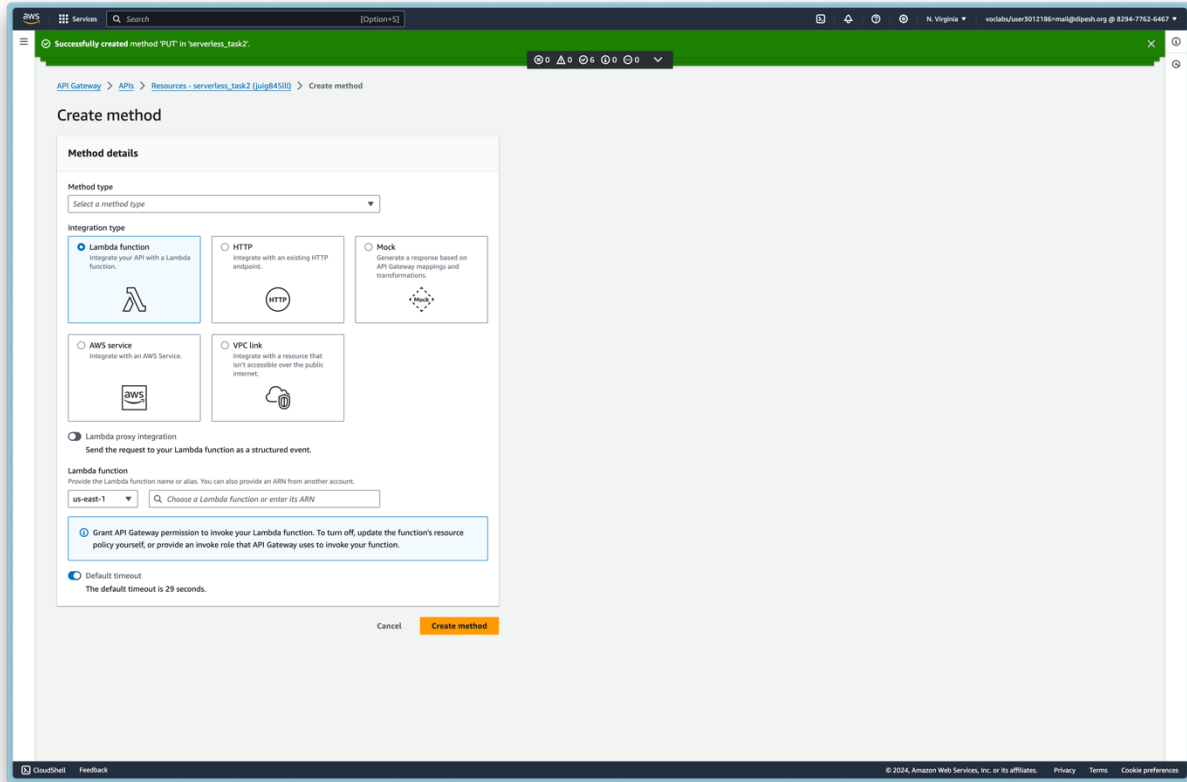
API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Regional

Cancel Create API

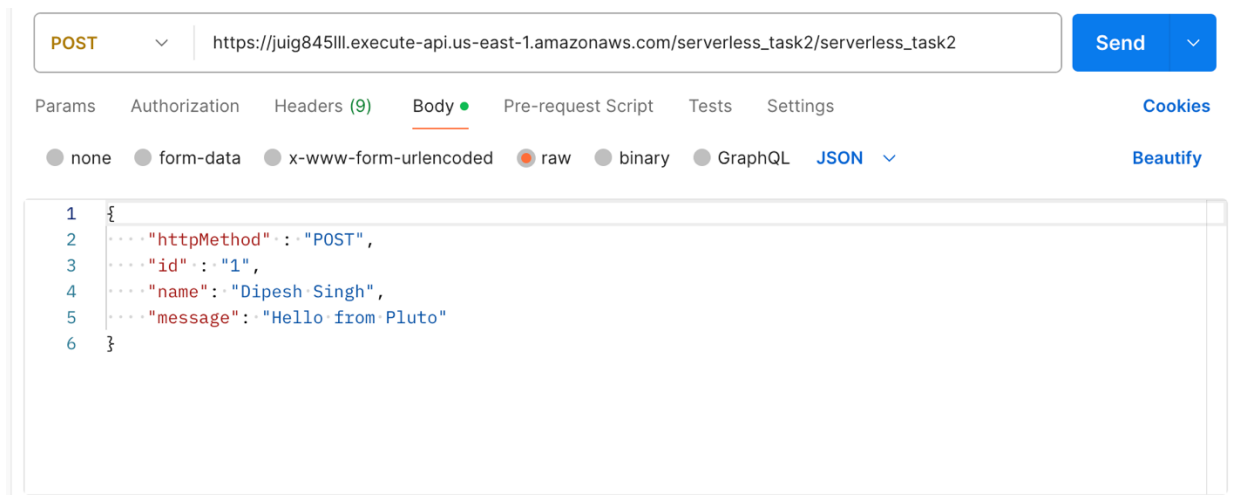
Create resources for API and enable CORS










Create GET POST PUT and DELETE method.



Use POSTMAN for testing API



It is updated in dynamodb table

Items returned (1)				Actions 	Create Item
<input type="checkbox"/>	id (String) 	message 	name 	< 1 >  	
<input type="checkbox"/>	1	Hello from Pluto	Dipesh Singh		