

## API Gateway

AWS API Gateway is a fully managed service that allows you to create, publish, and manage APIs at any scale. Key points:

- **Create RESTful and WebSocket APIs:** Supports both types of APIs.
- **Seamless Integration:** Works with AWS Lambda, EC2, and other AWS services.
- **Scalable:** Automatically handles traffic scaling.
- **Secure:** Supports IAM, Amazon Cognito, and custom authorizers for security.
- **Monitoring and Logging:** Integrates with CloudWatch for detailed monitoring and logging.
- **Cost-Effective:** Pay for the API calls and data transfer you use.

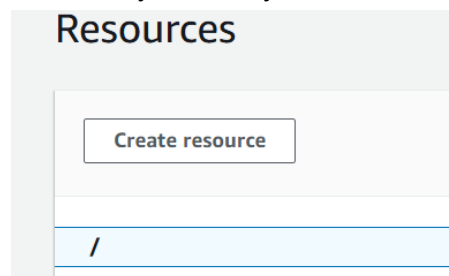
## Creating API and Configuration of API

### 1. Create New API:

- Search “API Gateway” in AWS console.
- Then after opening API Gateway Console click on “Create API” . On clicking it you will be given various options on what kind of API you want like HTTP API, Rest API, etc.
- I selected the Rest API. Choose **Build** under REST API.
  - **Create a new API:** Enter a name for your API
  - Optionally, enter a description and set up tags
  - Click **Create API**.

### 2. Define Resources and Methods:

- Now we will create resources for our API. To do this follow these steps:
- Click on your newly created API. Then select “Create resource” button



- Give a name to your resource. Then click the “create resource” button.
- Next we will create methods under our resource:
  - Click on the resource you created and then click on “Create Method”

- After that fill in the details:

Method Type: What kind of method do you want? POST, GET, PUT, etc

Integration Type: Select Lambda function in this

Lambda proxy integration: Switch ON this

Lambda Function: Select the Lambda function you created

Then click on the "Create Method" button.

### 3. Deploy the API:

- After Creating method click on Deploy API. Make sure you select the Resource under which you created your methods and then click Deploy API.

- Enter a Stage Name (e.g., **dev**, **test**, or **prod**).
- Click **Deploy**.
- Now after doing this you can see the **Invoke URL**. This URL will be used to test your API with POSTMAN.

### 4. Before testing your API. You need to do one more thing. You need to grant **API Gateway permissions to invoke your Lambda function**. To do this follow these steps:

- Go to your "Lambda function" console.
- In that select the "Configuration" tab. In this select "Permissions"

Code Test Monitor **Configuration** Aliases Versions

General configuration

Triggers

**Permissions**

Destinations

**Execution role** [Refresh](#) [Edit](#) [View role document](#)

Role name  
[Burn\\_data\\_Lambda\\_Access](#)

**Resource summary**

- Scroll down to “Resource-based policy statements”. In that click “Add permissions”.

**Resource-based policy statements (13)** [Info](#)

[Refresh](#) [View policy](#) [Edit](#) [Delete](#) [Add permissions](#)

A resource-based policy lets you grant permissions to other AWS accounts or services on a per-resource basis.

	Statement ID	Principal	PrincipalOrgID	Conditions
<input type="radio"/>	<a href="#">ef4241fa-25b5-5fd6-88ac-04ffb5b057a5</a>	apigateway.amazonaws.com	-	ArnLike
<input type="radio"/>	<a href="#">c9ca0222-f97b-5fe8-8ab1-a300d4bf92e2</a>	apigateway.amazonaws.com	-	ArnLike
<input type="radio"/>	<a href="#">ReadData</a>	apigateway.amazonaws.com	-	ArnLike

- In that add the details:

**Edit policy statement**

☐ AWS account  
Grant permissions to another AWS account, user, or role.

☒ AWS service  
Grant permissions to another AWS service.

☐ Function URL  
Grant permissions to invoke your function through the function URL.

**Service**  
The AWS service to grant permissions to.

**Statement ID**  
Enter a unique statement ID to differentiate this statement within the policy.

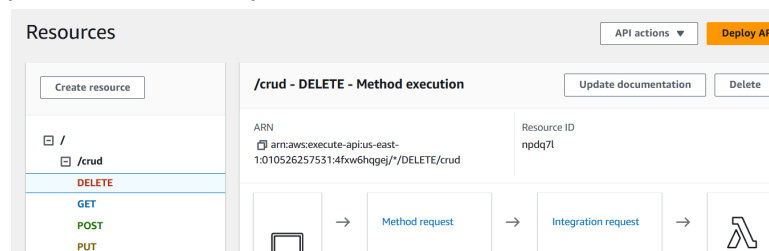
**Principal**  
The service principal for this AWS service. [Learn more](#)

**Source ARN**  
The ARN for a resource. Find the ARN in the related service console.

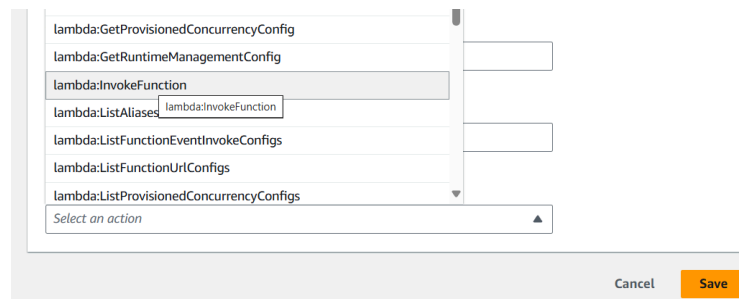
**Action**  
Choose an action to allow.

- Select “AWS service” in the options
- Service: Select “API Gateway” in this.

- Statement ID: Name the permission whatever you want.
- Source ARN: In this select the ARN from your API. Just select the method and you can ARN. Copy that ARN here.



- Action: In this select “lambda:InvokeFunction”.



- After filling this click on “Save” button
- Note: You need to create this permission for every Method you created in the API. Like for Delete Method one permission, for POST Method one permission and so on.

## 5. Testing the API

- Open POSTMAN
- Create New Request
- Configure Request for CRUD Operations:

**GET Request:** Set the method to **GET**.

Enter the Invoke URL with a query parameter:

<https://your-api-id.execute-api.region.amazonaws.com/dev/crud?id=1>

So, in this way you can Create API and integrate your API with Lambda Functions. Now these API endpoints you created can be used later to perform CRUD operations with the website which you will create later.