Build API Gateway with Lambda Integration

Level: **Intermediate**

AWS Lambda Amazon Web Services Amazon API Gateway

English

Required Points	10
Lab Duration	00:45:00
Average Start time	Less than a minute

Start Guided Lab →

Lab Overview Lab Steps

Cloud Architect, Cloud DevOps Engineer

Networking, Serverless

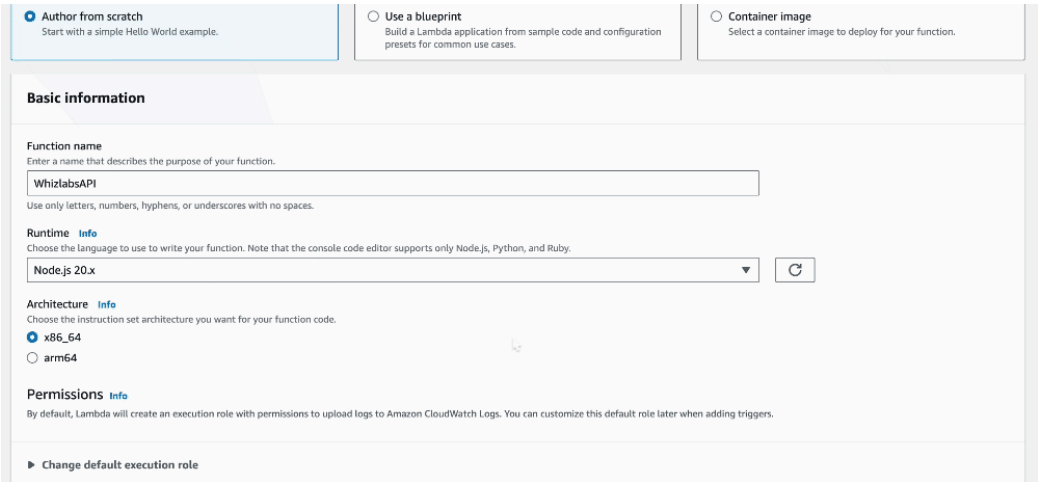
Lab Steps

Task 1: Sign in to AWS Management Console

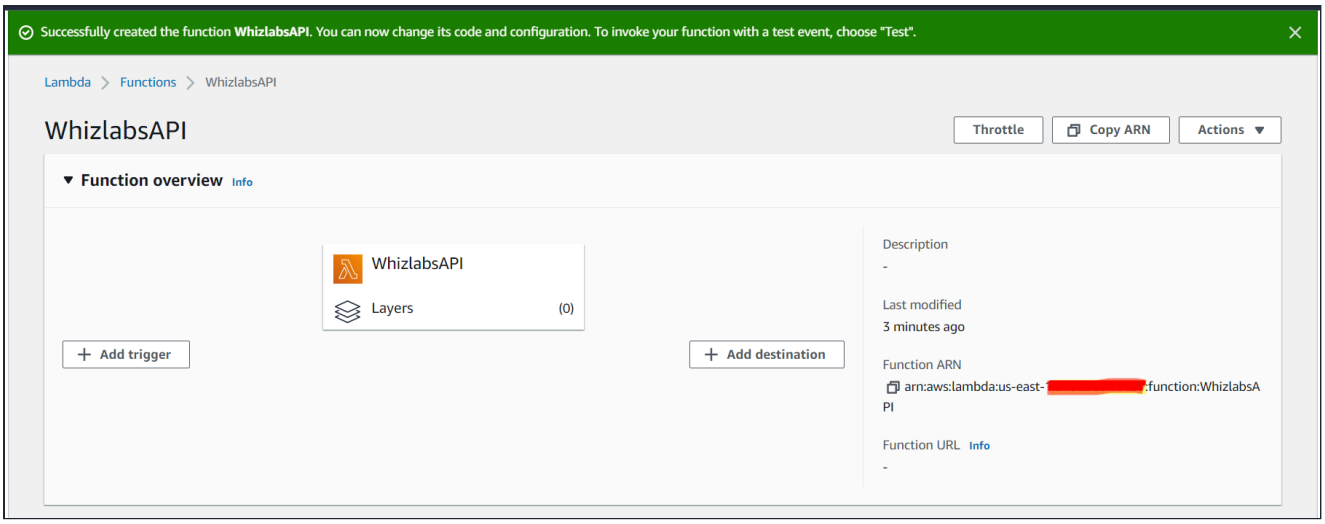
1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
    - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
    - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US (N. Virginia) us-east-1.**

Task 2: Create a Lambda Function

1. Navigate to the **services** menu at the top, then click on **Lambda** under the **Compute** section.
2. Click on the **Create a function** button.
    - Choose: **Author from scratch**.
    - Function name: Enter ***WhizlabsAPI***
    - Runtime: Select **Python 3.12**
    - Role: In the permissions section, choose **create a new role from AWS policy templates** under **change default execution role**.
    - Enter the Role name as ***WhizlabsAPI***, Choose policy template as **Basic Lambda @ Edge Permission(For CloudFront Trigger)**
        - Click on **Create function**

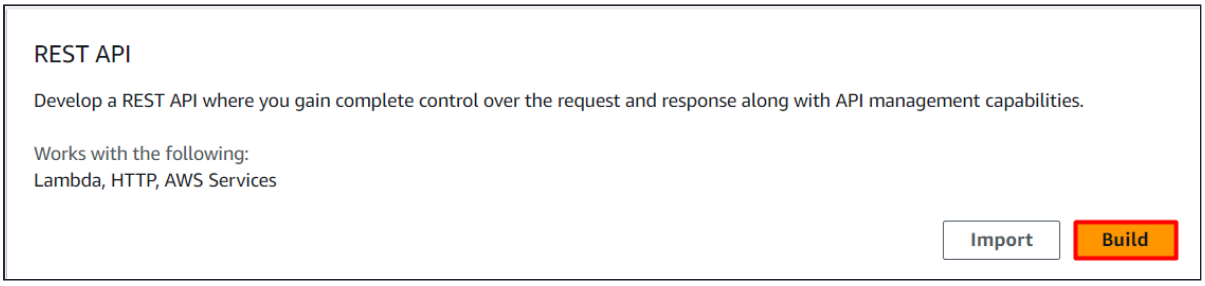


3. Once the Lambda Function is created successfully, it will look like the screenshot below:



Task 3: Create an API

1. Navigate to the **services** menu at the top, then click on **API Gateway** in the **Network and Content Delivery** section.
2. Click on **Get Started**. (If gets started and is not visible), then click on **Build** in REST API and select Protocol as **REST**. (Close the pop up message for Create your first API,if it is present and ignore the error warning).



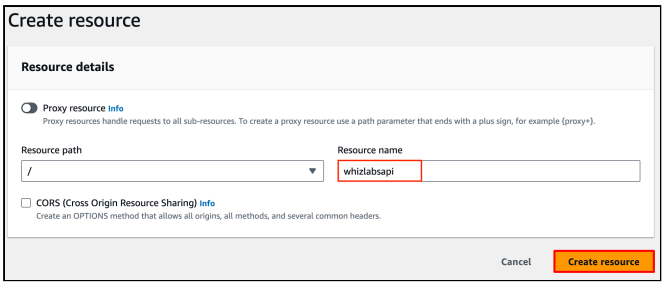
3. Choose to create a new API as **New API.** Under settings, choose the API name ***WhizlabAPI.*** Leave other options as default and click on **Create API**

Task 4: Creating a Resource

1. Once the API is created, select the API.
2. Select **Create Resource** in actions.

    - Resource Name: Enter ***whizlabsapi***

3. Enter the resource name and click on **Create Resource**

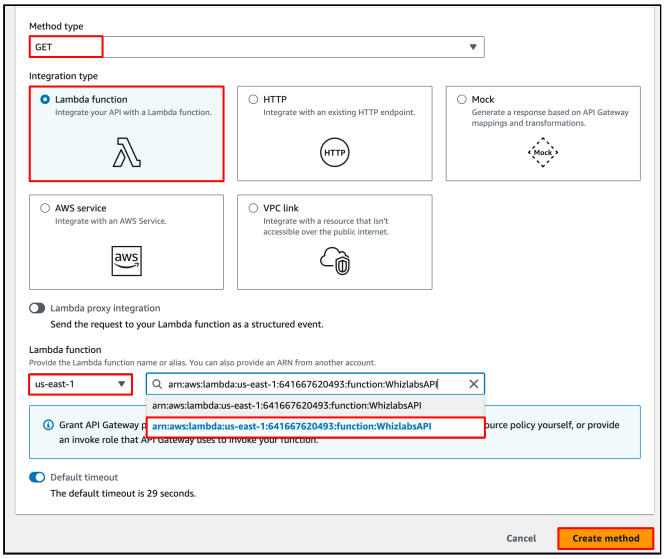


Task 5: Creating a Method

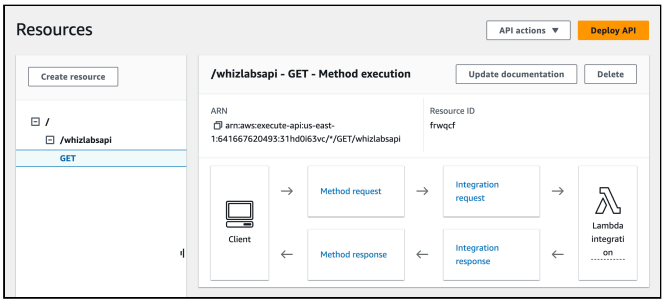
1. Once you create a resource, click on **Create Method**. Select **Get** in the drop-down list of **Method Type**.

2. Select the Integration Type as **Lambda function.**

3. Enter the Lambda Function as ***WhizlabsAPI*** and choose the **us-east-1** region. Click on **Create Method**

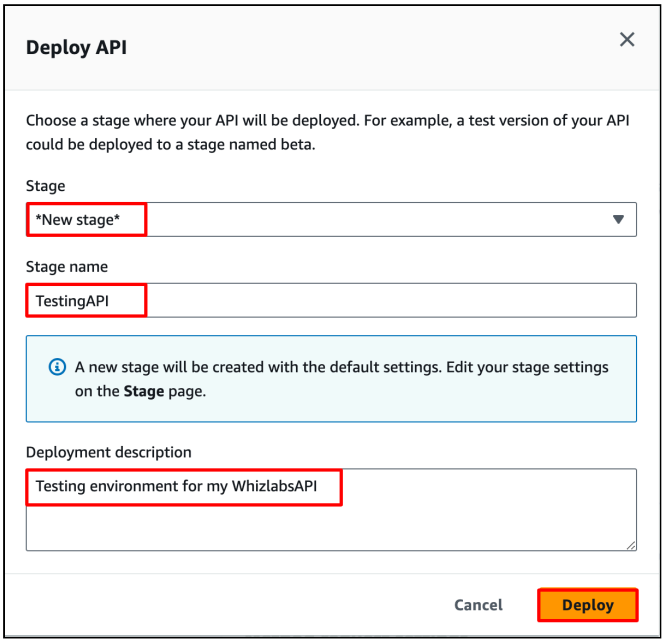


4. Once the method has been created, your screen will look similar to the screenshot below:

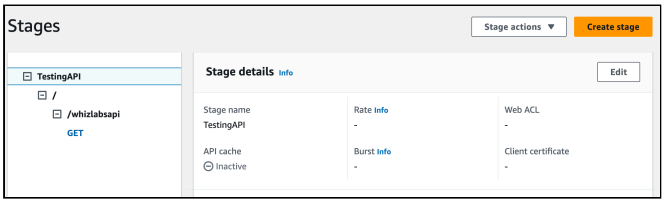


Task 6: Deploy the API

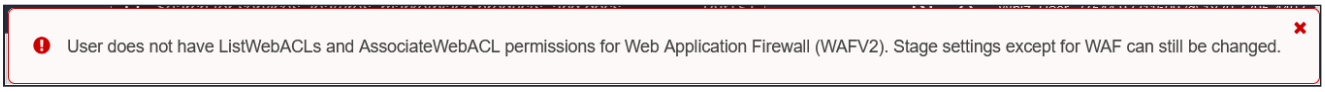
1. Once the resource and the method have been created successfully, you can deploy the API.
2. Click on **Deploy API**.
3. Select the Deployment Stage in the drop-down as **New Stage**.
4. Enter Stage Name: ***TestingAPI*** and Stage description as **Testing environment for my WhizlabsAPI.**
5. Click on **Deploy.**



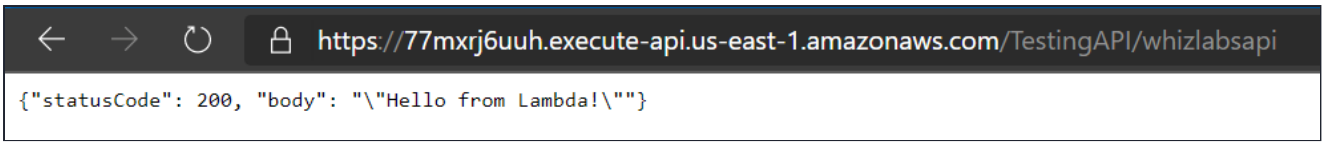
6. Once the API has been deployed, navigate to **Stages**. You will be able to see the following:



7. If you are getting the below permission error, ignore it. It is not required for this lab.



8. Click on the **Invoke URL** you find under **TestingAPI Stage Editor** in the new tab of your browser to make a GET request.
9. Now add ***/whizlabsapi*** at the end of deploy URL and hit Enter.
10. You will receive the GET request from the API. Here's an example:



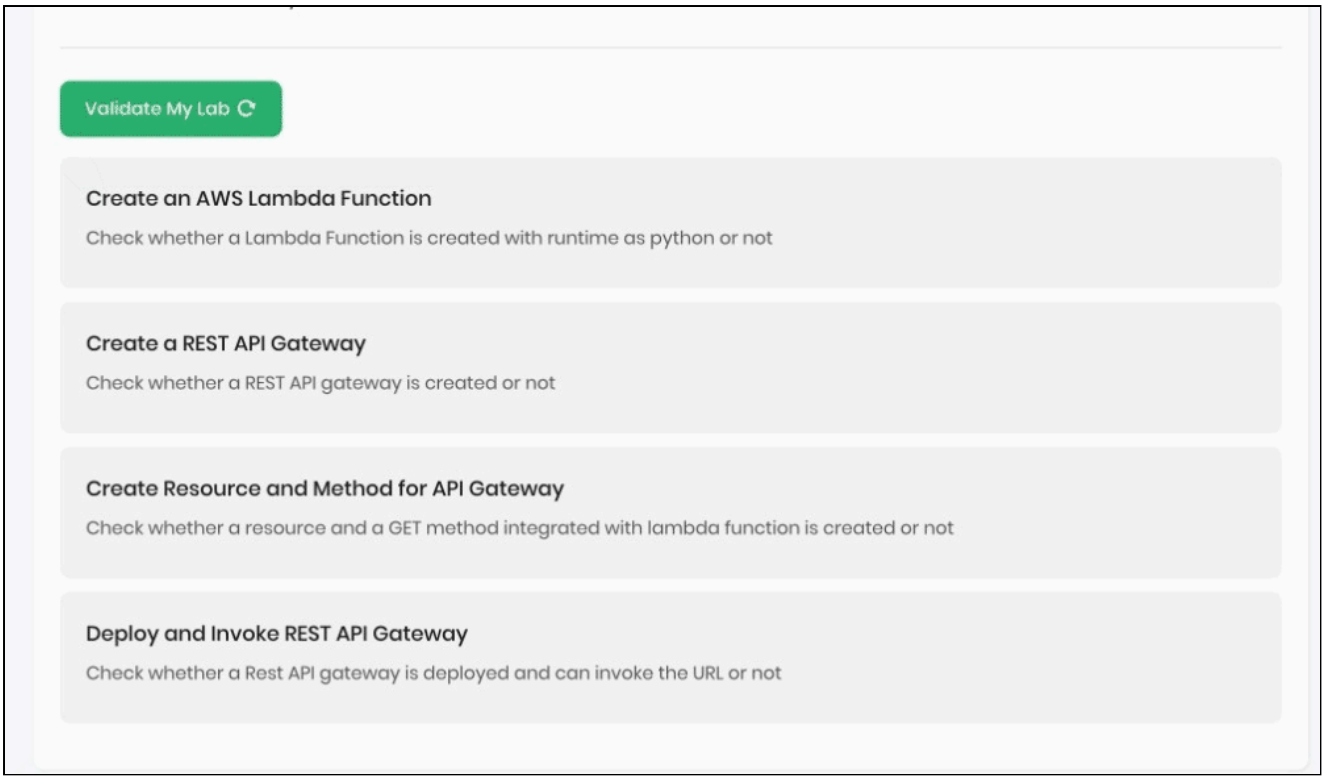
{"statusCode": 200, "body": "\"Hello from Lambda!\""}

Do you know?

1. API Gateway is a fully managed AWS service that enables developers to create, publish, and manage APIs for their applications. It acts as a front-door to your backend services, allowing you to securely expose your APIs to clients, handle authentication and authorization, and apply various transformations and optimizations.
2. AWS Lambda is a serverless compute service provided by Amazon Web Services. It allows developers to write and execute code without provisioning or managing servers. A Lambda function is a piece of code that performs a specific task or handles a specific API request. It can be written in various programming languages and is triggered by events, such as an API request.
3. Lambda integration in API Gateway refers to the process of connecting an API endpoint with a Lambda function. It enables API Gateway to invoke the Lambda function in response to incoming API requests, allowing the function to process the request, perform business logic, and generate a response.

Task 7: Validation Test

1. Once the lab steps are completed, please click on the **validation button** on the right side panel.
2. This will validate the resources in the AWS account and shows you whether you have completed this lab successfully or not.
3. Sample output :



Completion and Conclusion

1. You have successfully created a Lambda function.
2. You have successfully created the API.
3. You have successfully created the API Resource and Method.
4. You have successfully tested the API.

End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs dashboard.