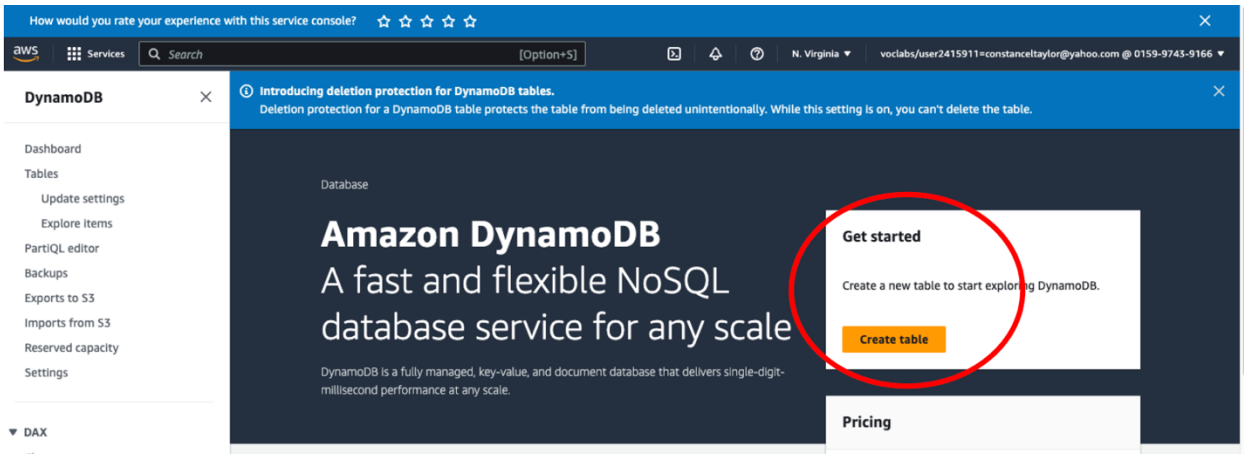
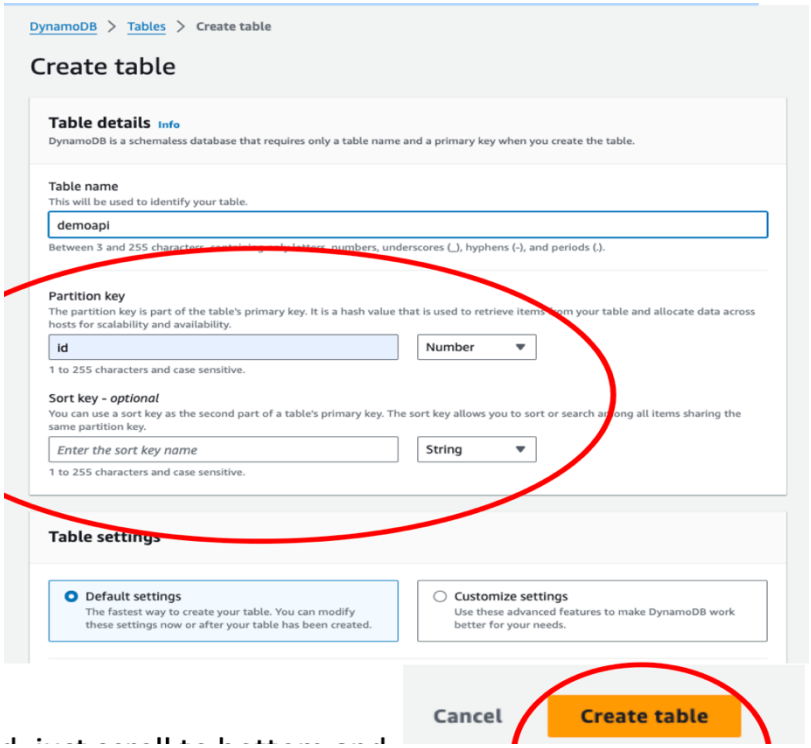
CPSC 3720: Creating an API in AWS Homework Assignment

This assignment should reinforce the concepts reviewed in class and assist you with Sprints 2- 4. For this homework you will create an example API in AWS using the following AWS Services provided in the AWS Console-→Services: 1) Lambda, 2) API Gateway, and 3) DynamoDB. You will test your API in Postman.



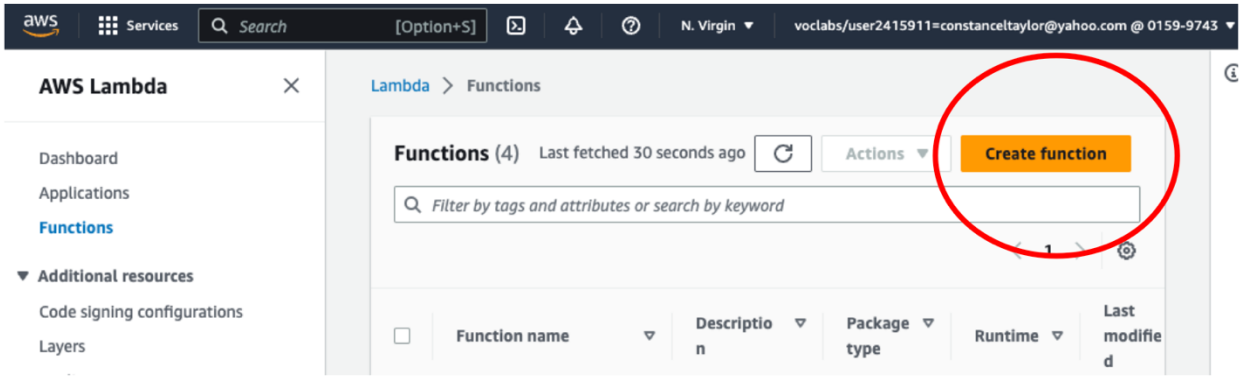
- Name it “demoapi” and give it a partition key called “id” with type Number



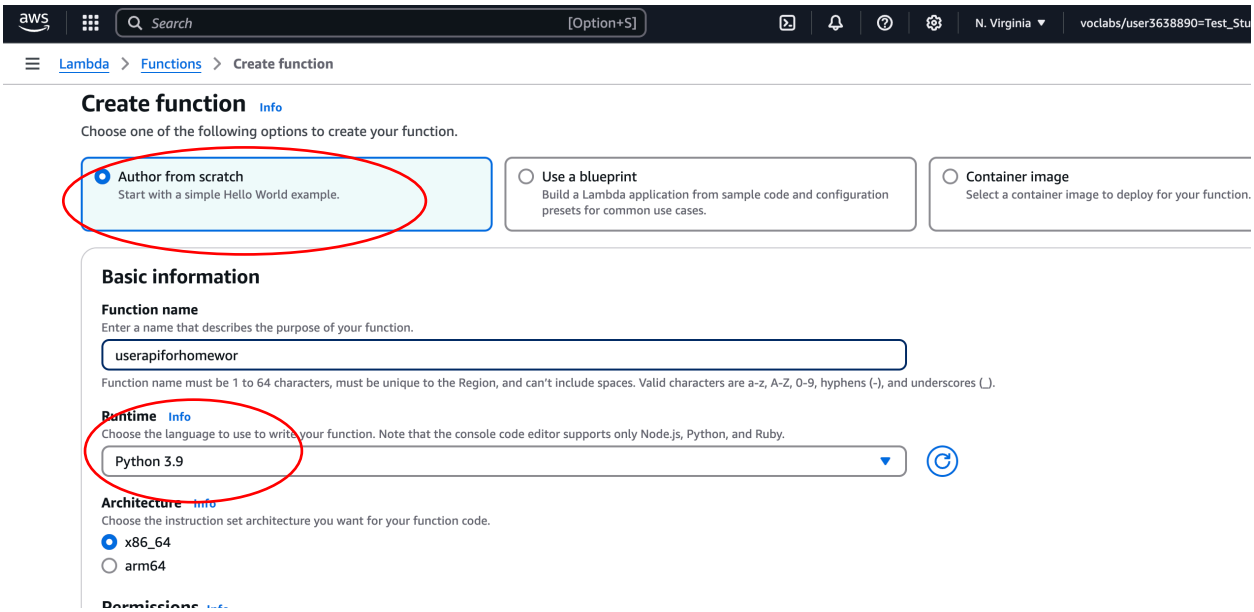
- No other changes need; just scroll to bottom and create table

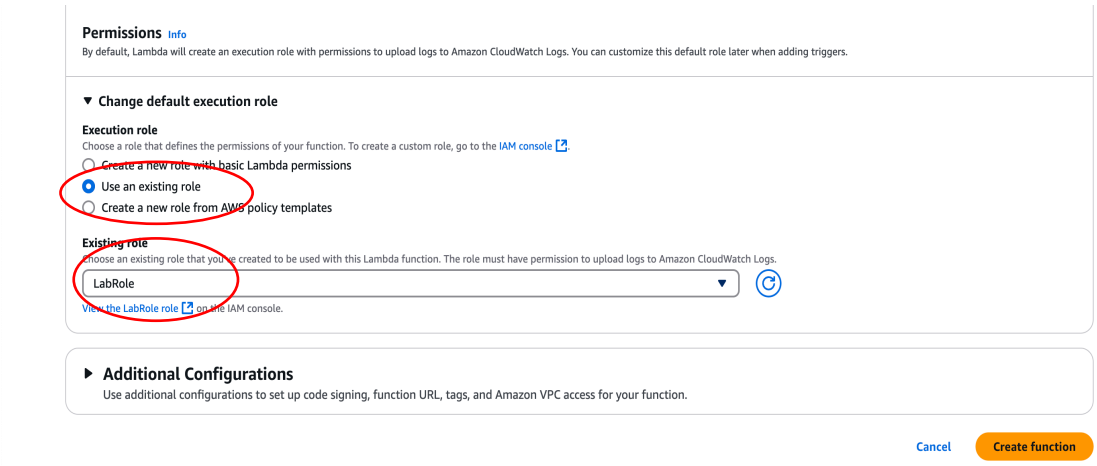
2. **Create a Lambda function (you can use either python or node to build the application)**

Lambda function with Python code:



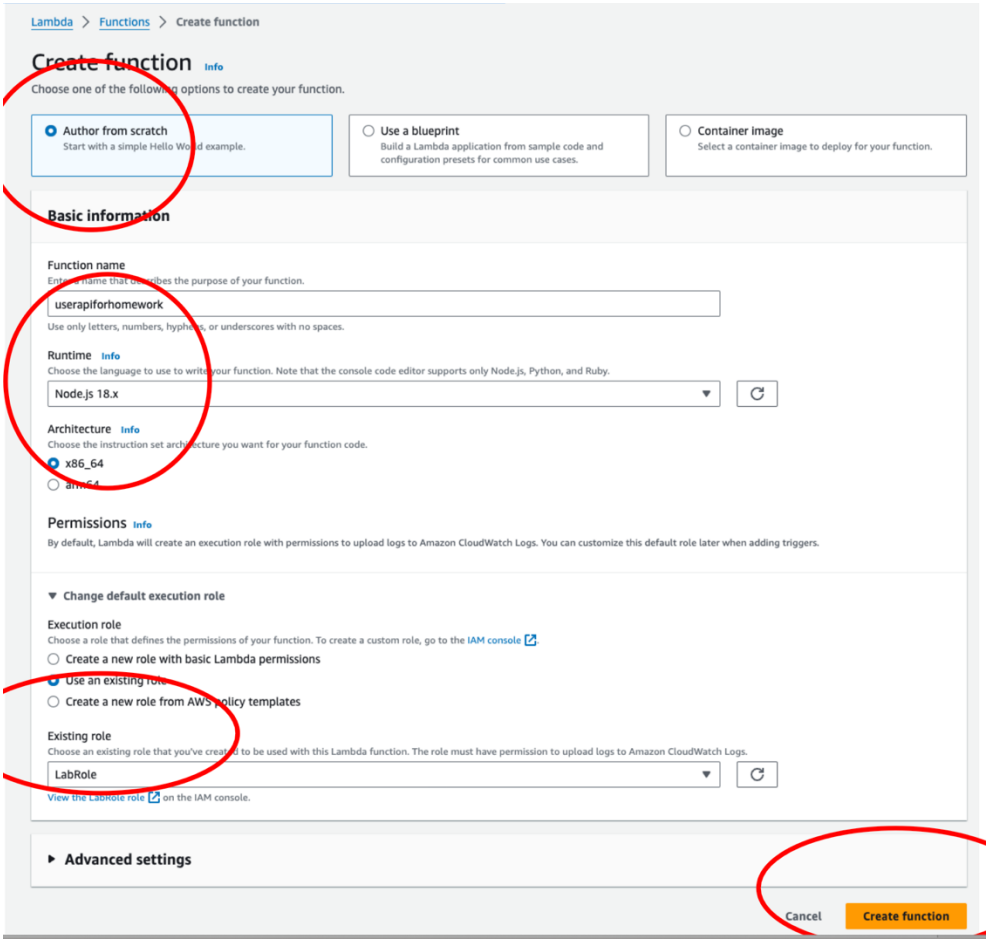
- Author for Scratch Python runtime 3.9**x version**!
- Select Use existing Role and type in LabRole





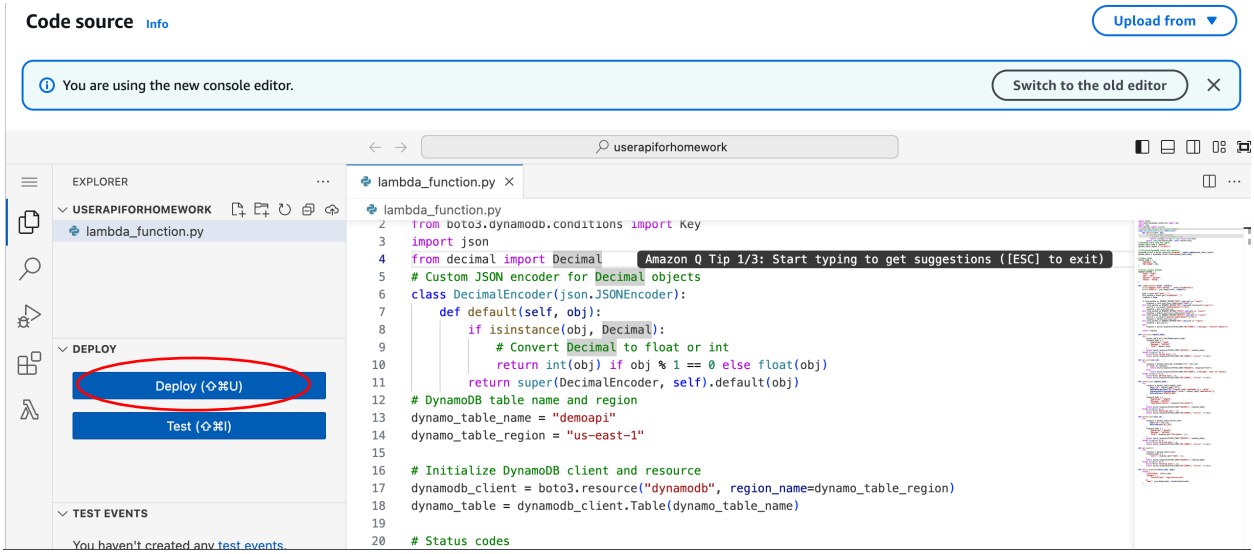
(OR) Lambda function with Python code:

- Author for Scratch Node js runtime **18.x version**!
- Select Use existing Role and type in LabRole

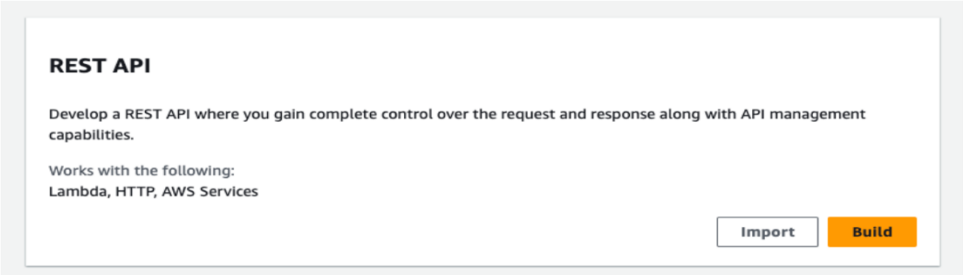


• Scroll to the bottom and select **Create Function**.

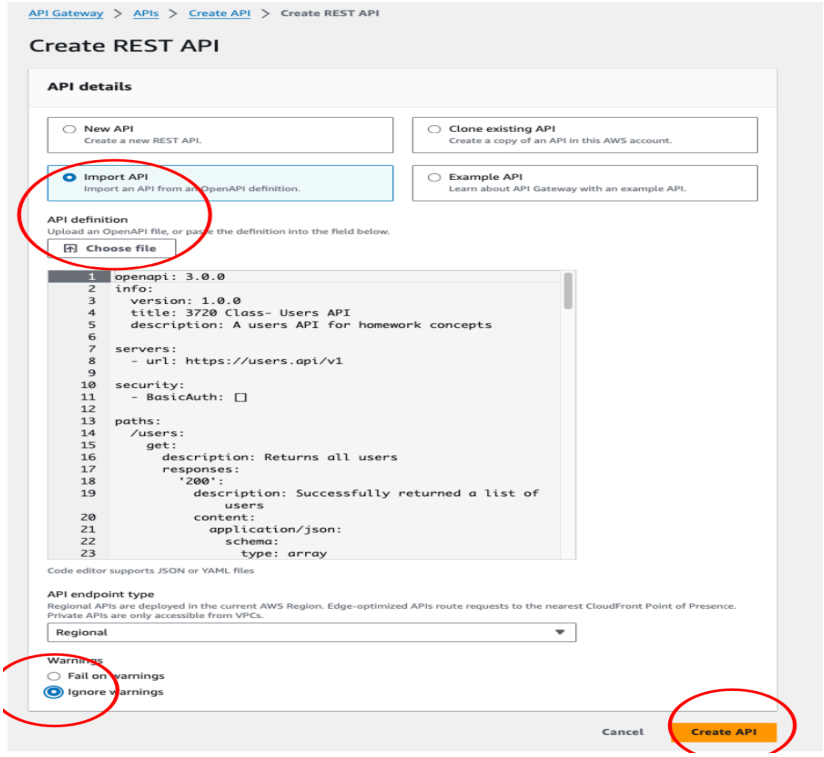
Copy in the code from index_python.py or indexv18node file provided with this assignment into the **code tab** in Lambda and then **deploy** the code. Anytime you change the code it will require a redeploy.



Create API with API Gateway a new REST API (not private)

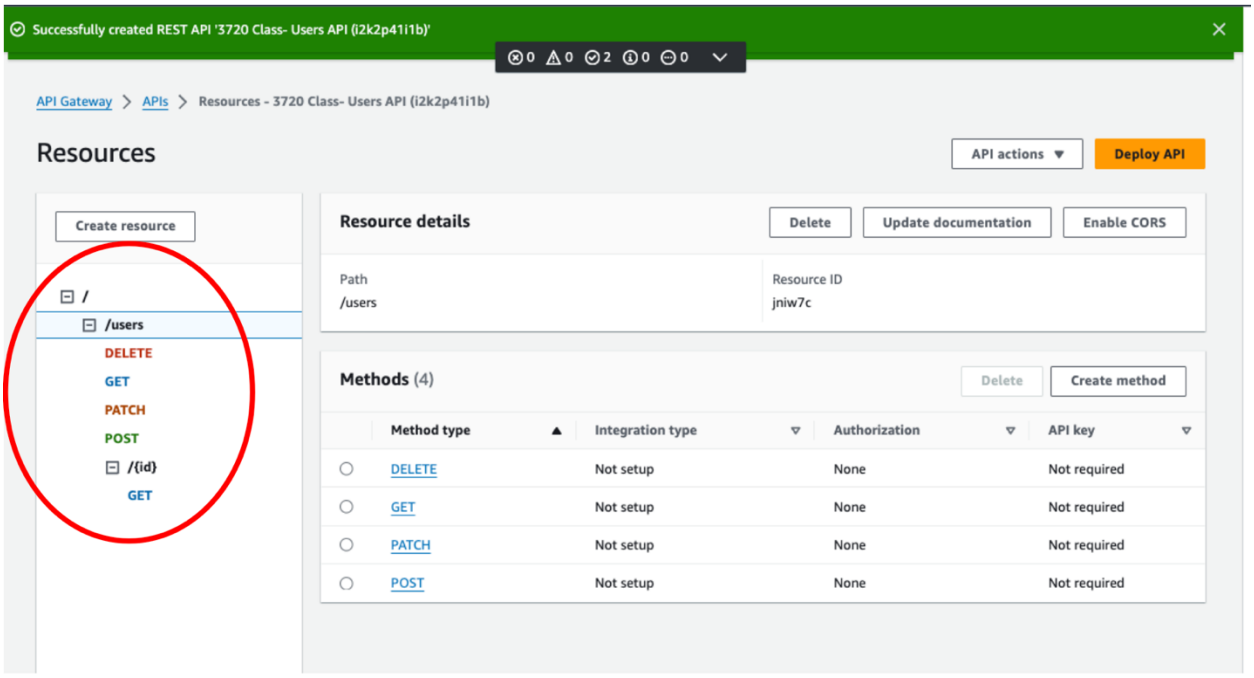


- Select Import an API from a definition and use the AWSHomework.yaml file provided with the assignment to import.
- Also selection – Ignore Warnings on the bottom of the window

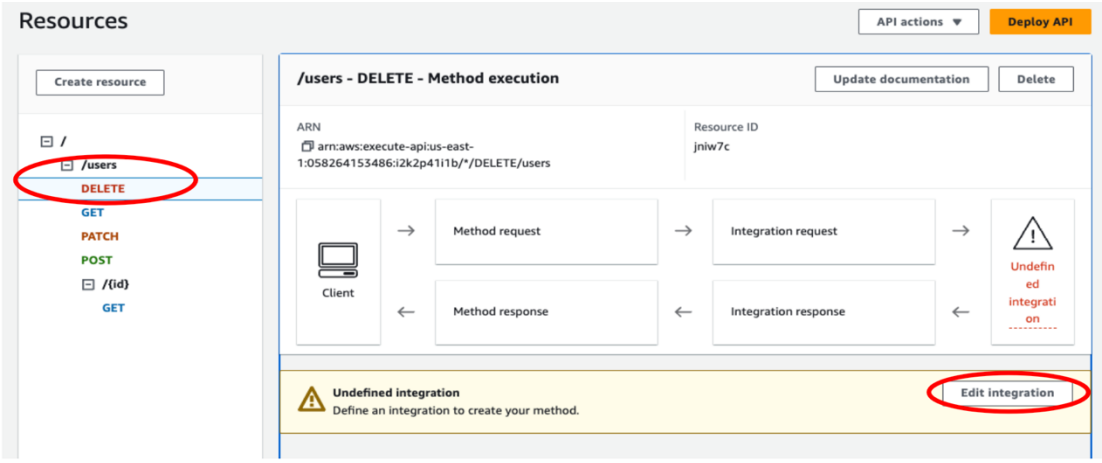


- Create API on the bottom.

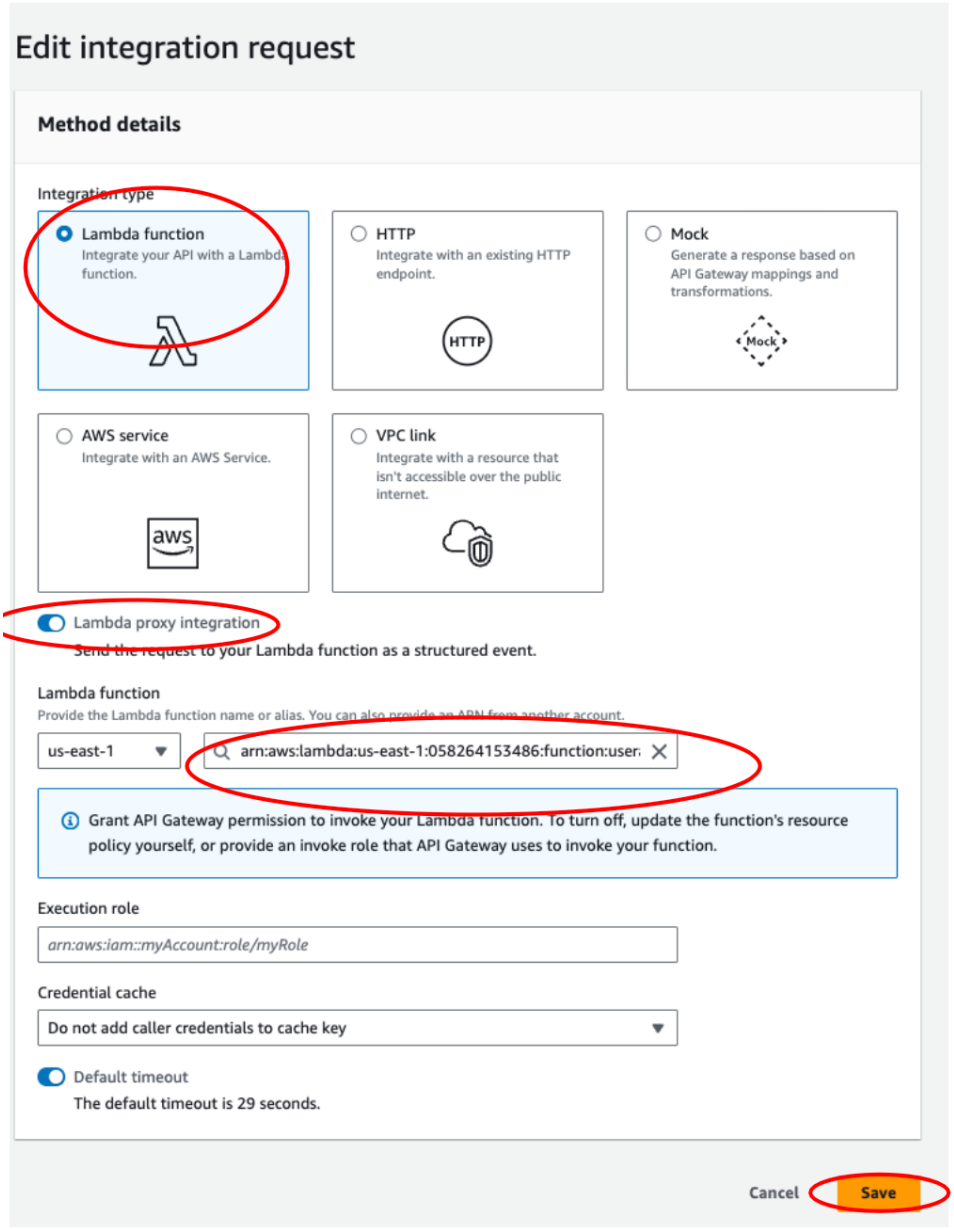
Now you have all the Resources and methods from the file ready to connect to the Lambda functions!



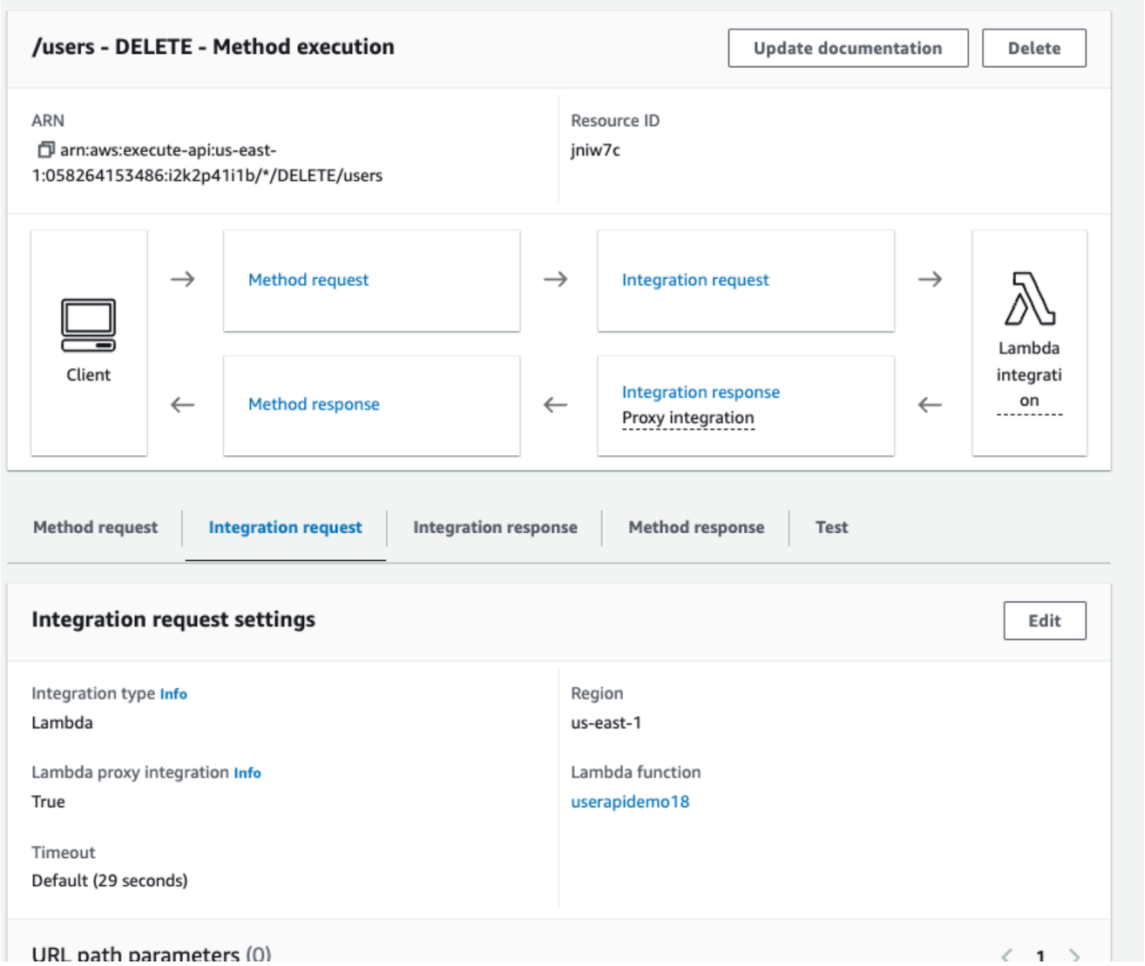
• For each method of your two path (aka resources) you will need to connect them to the Lambda function. For example, to connect the /users Delete method to the Lambda function, you select Delete on the left Nav, and you will see the following screen:



• Select **Edit Integration** and you will see the following screen which you will choose Lamda Function Integration, Lambda Proxy Integration, and then choose the Lambda function you completed in Step 2 previously:

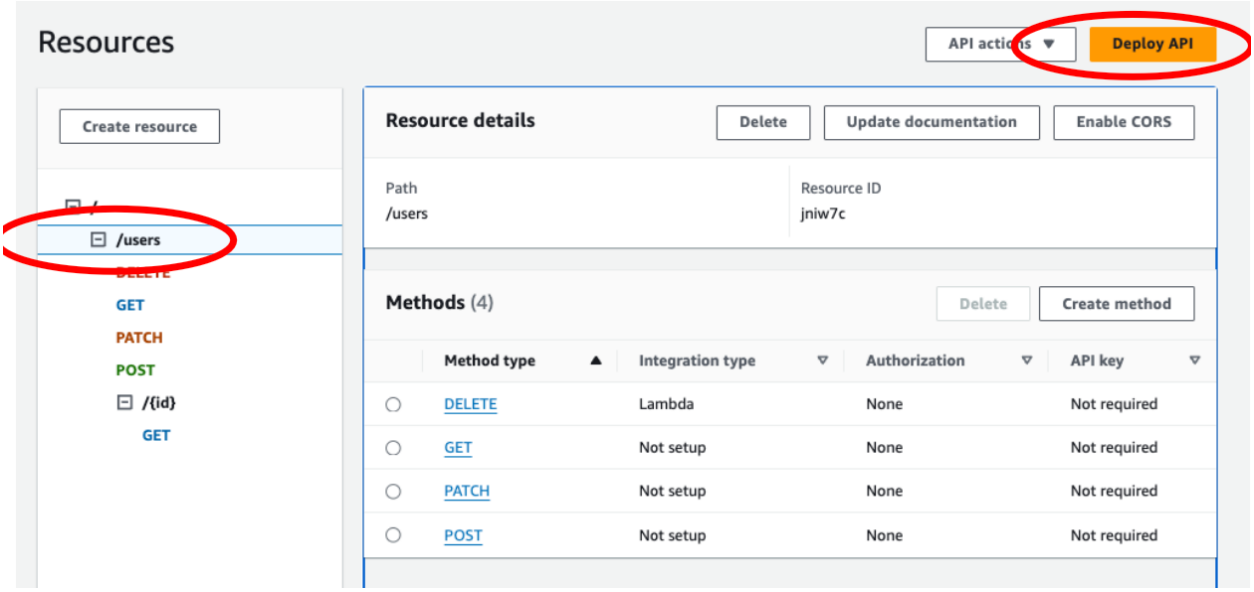


• Select Save when and you should be presented a screen similar to the following:

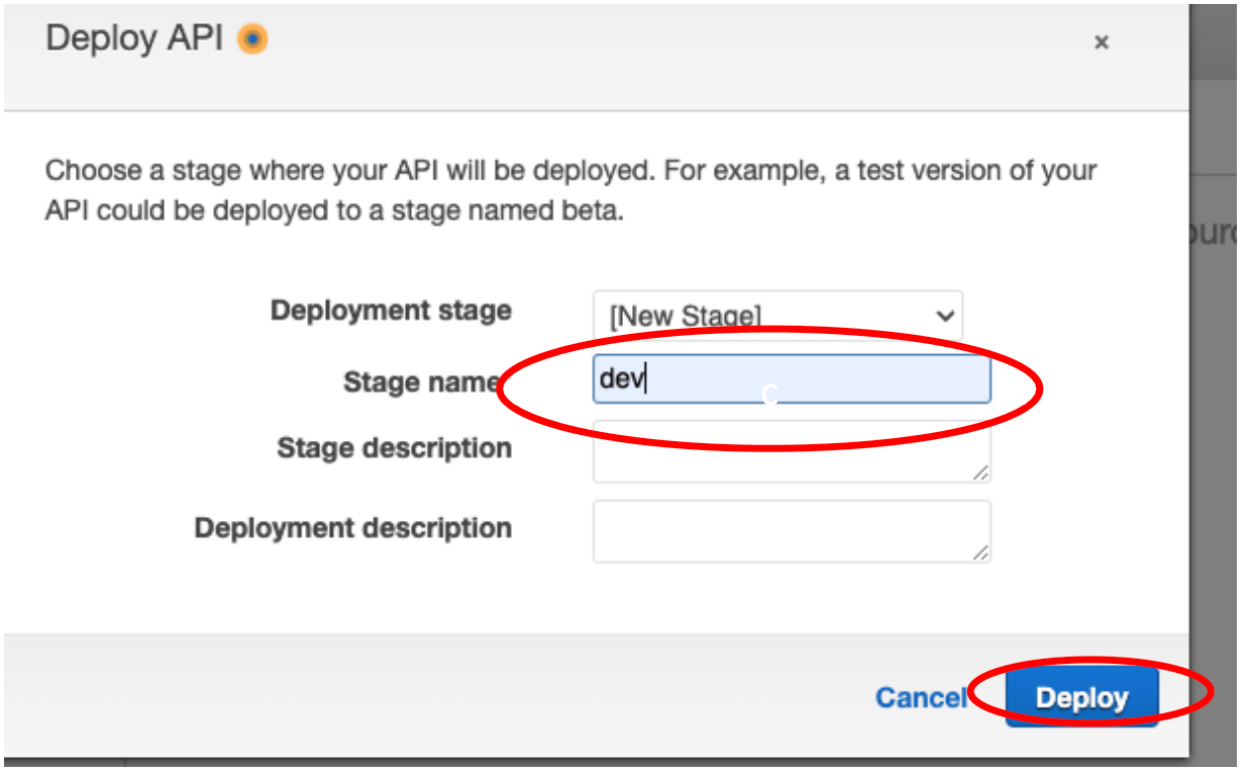


- **Repeat the lambda integration for all methods** for both paths: *users* and *users/{id}*

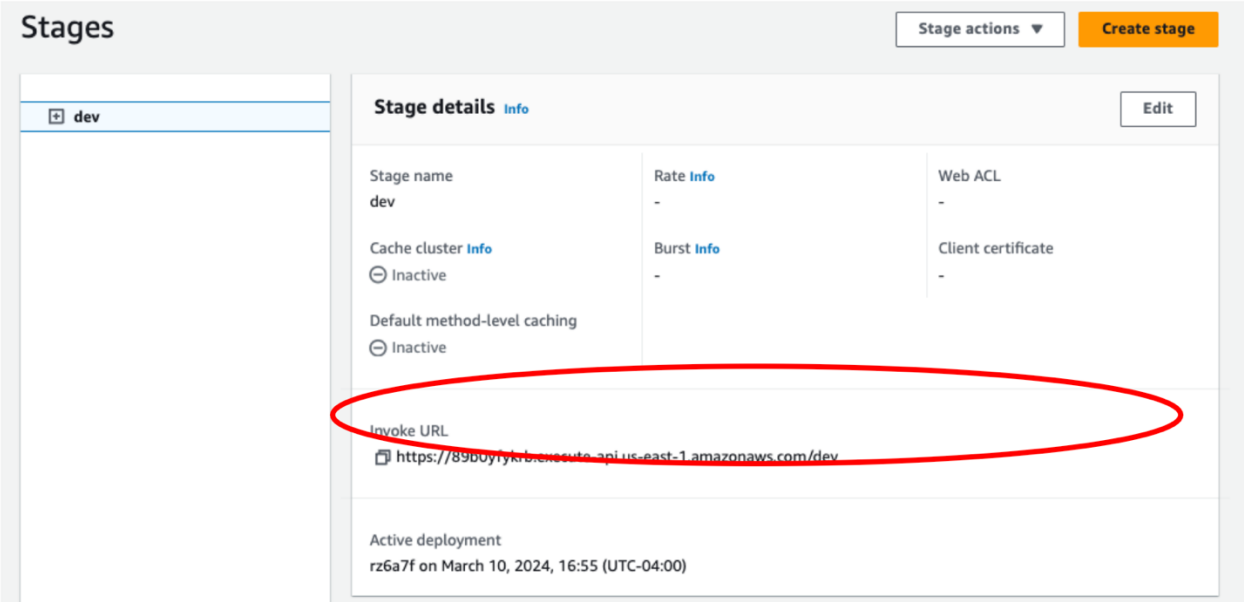
- Now it's time to deploy the API so you can test it out! Once you have completed your integration to the Lambda function for all 5 methods, go select the /users root of your API and Deploy API.



- Create a new Stage environment for your deployment and call it dev, then Deploy:



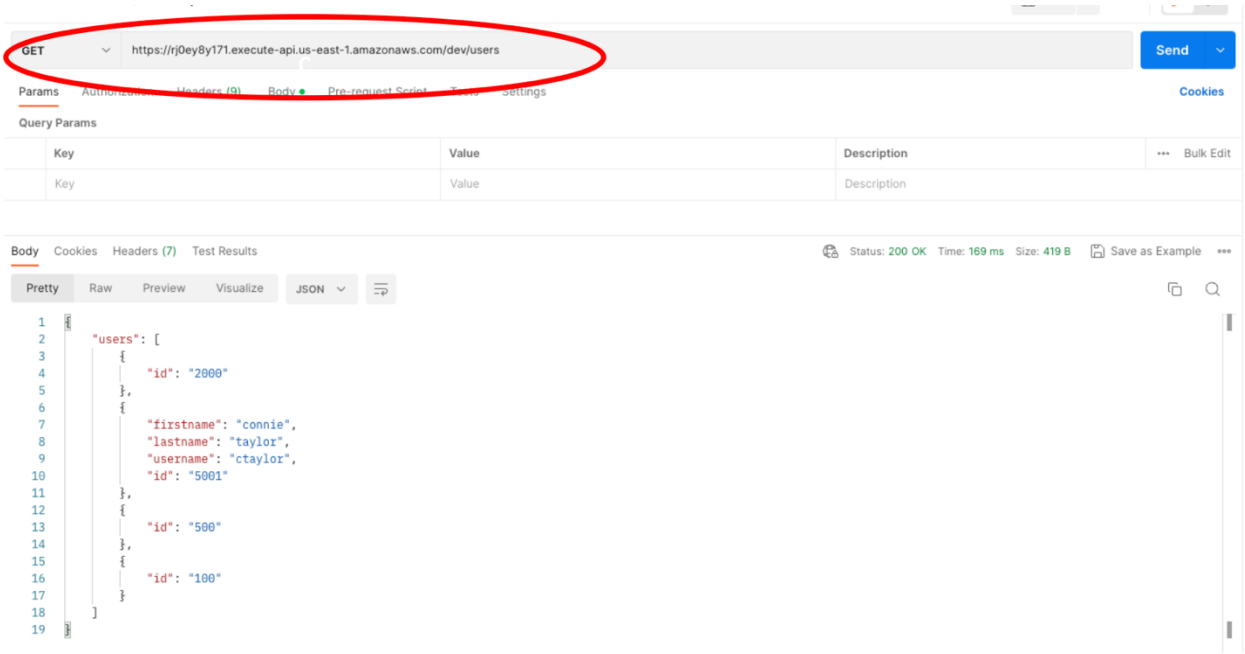
- The next screen will show you the URL that was created for your API:



NOW YOU HAVE A WORKING API IN AWS = CONGRATULATIONS!!!!

4. Test your API in Postman

- Let's make sure your API actually works!
- Use the API definition that was imported into API Gateway to create a collection in Postman for testing your API
- You can use a variable to rename the URL to the actual URL provided by AWS.
- Here is an example running the /users GET to get all users (I am not using an environment variable in this case so you can see the full URL):



• **You should run all five methods and show output**. For the POST and PATCH you will need a request Body. NOTE: You can add more data to the Dynamo DB by using the POST method on /users or you can directly add data in Dynamo DB.