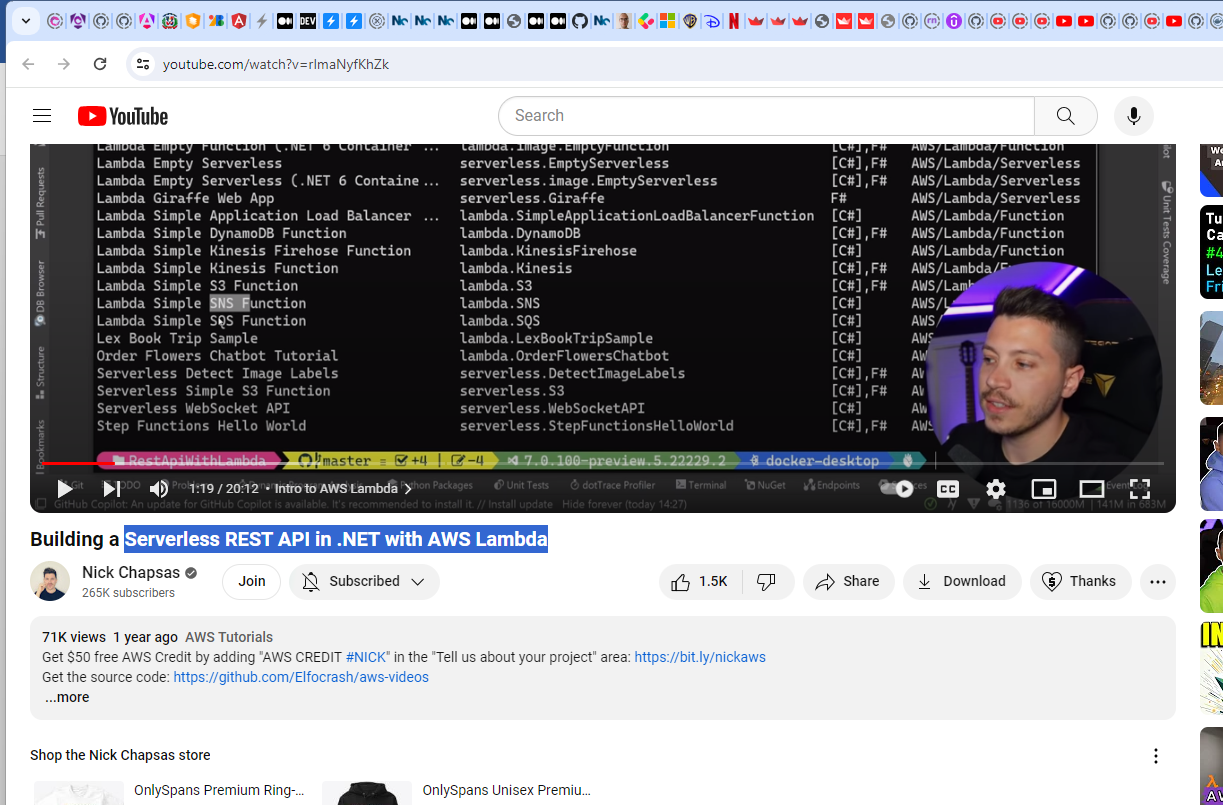
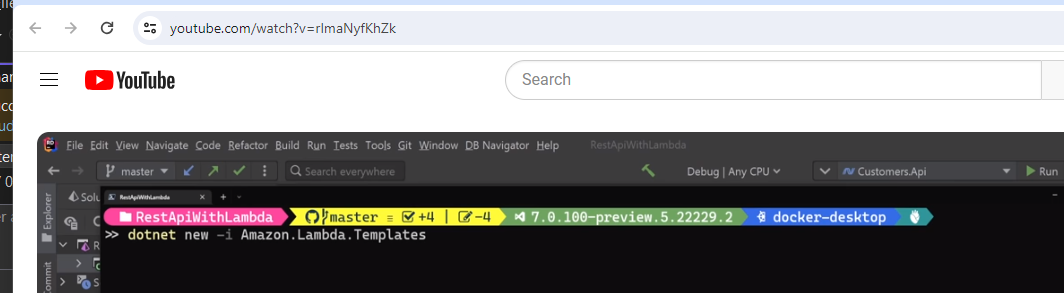
# Notes to Serverless REST API in .NET with AWS Lambda

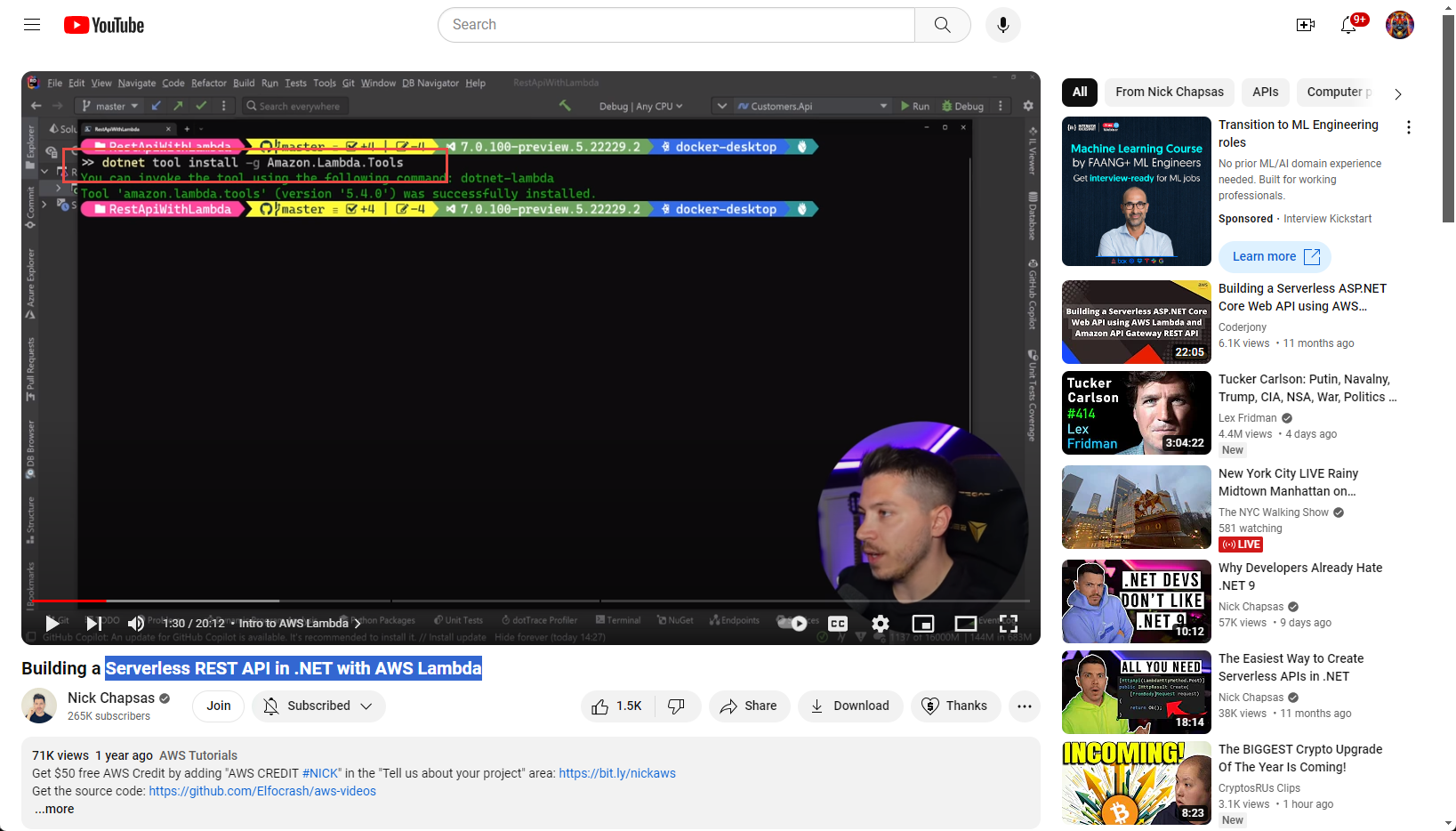
Notes from this video:

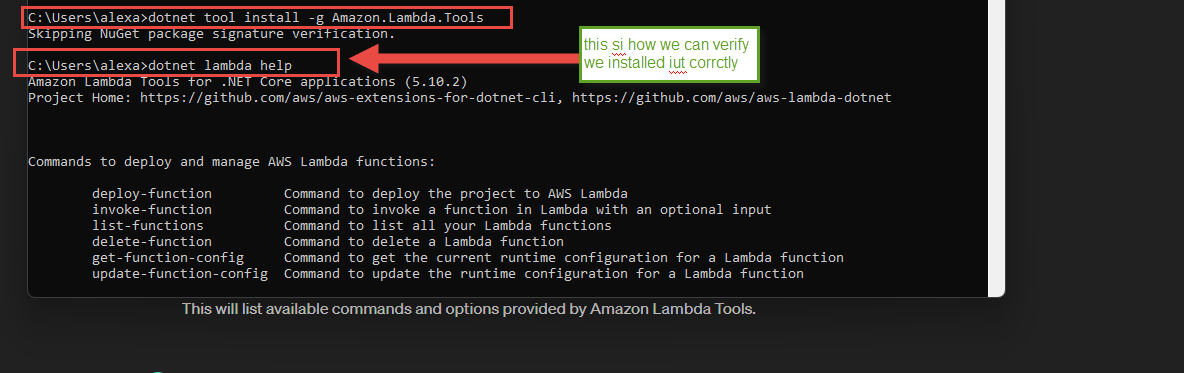


To see all these template in visual studio we need to install:



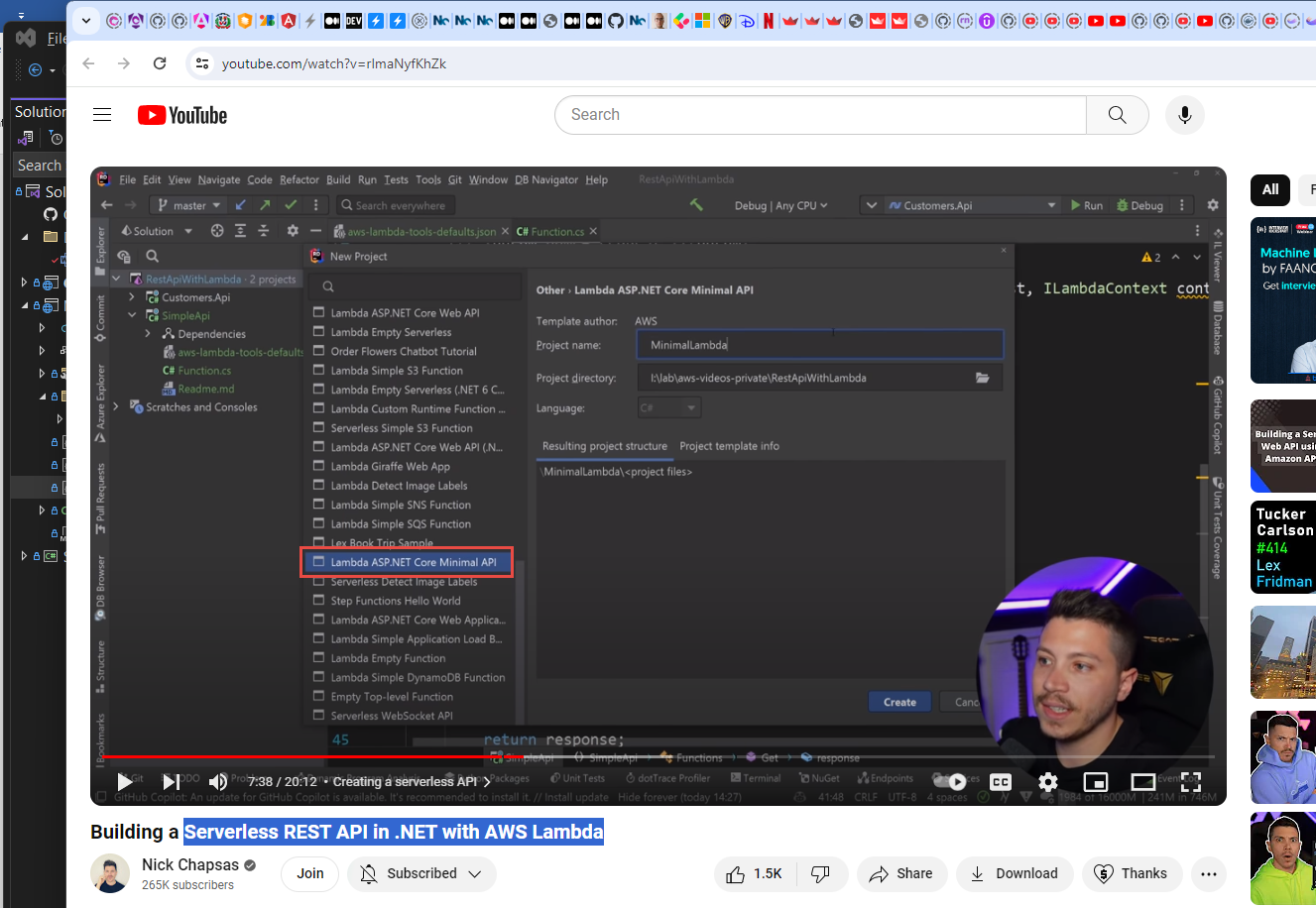
And also this



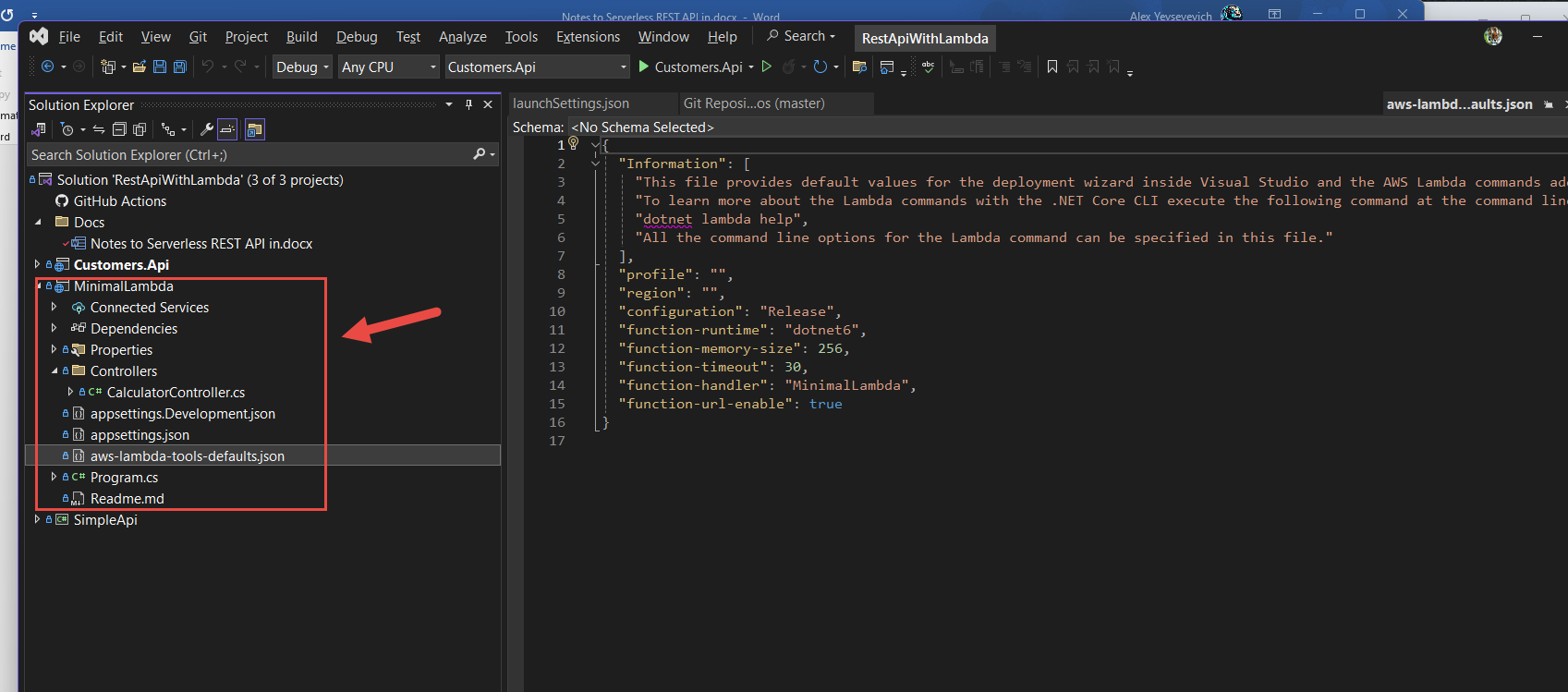


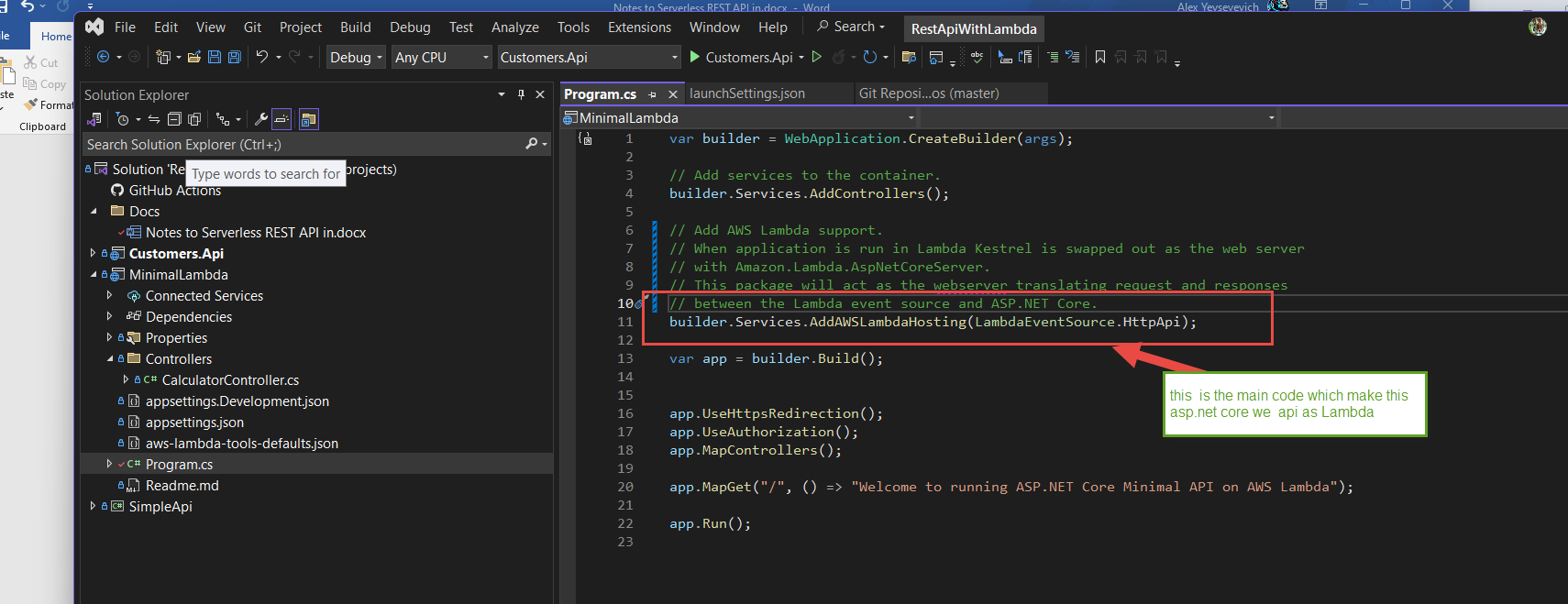
I think i have all of there with AWS tool install on my machine

Now let us create this project using this template

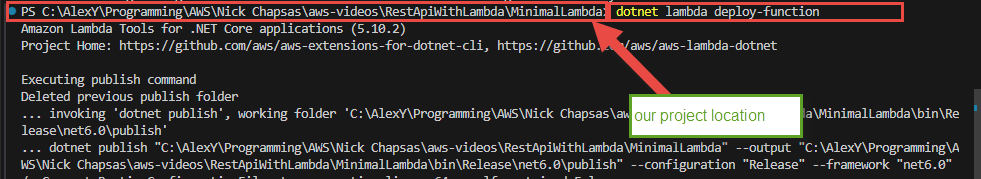


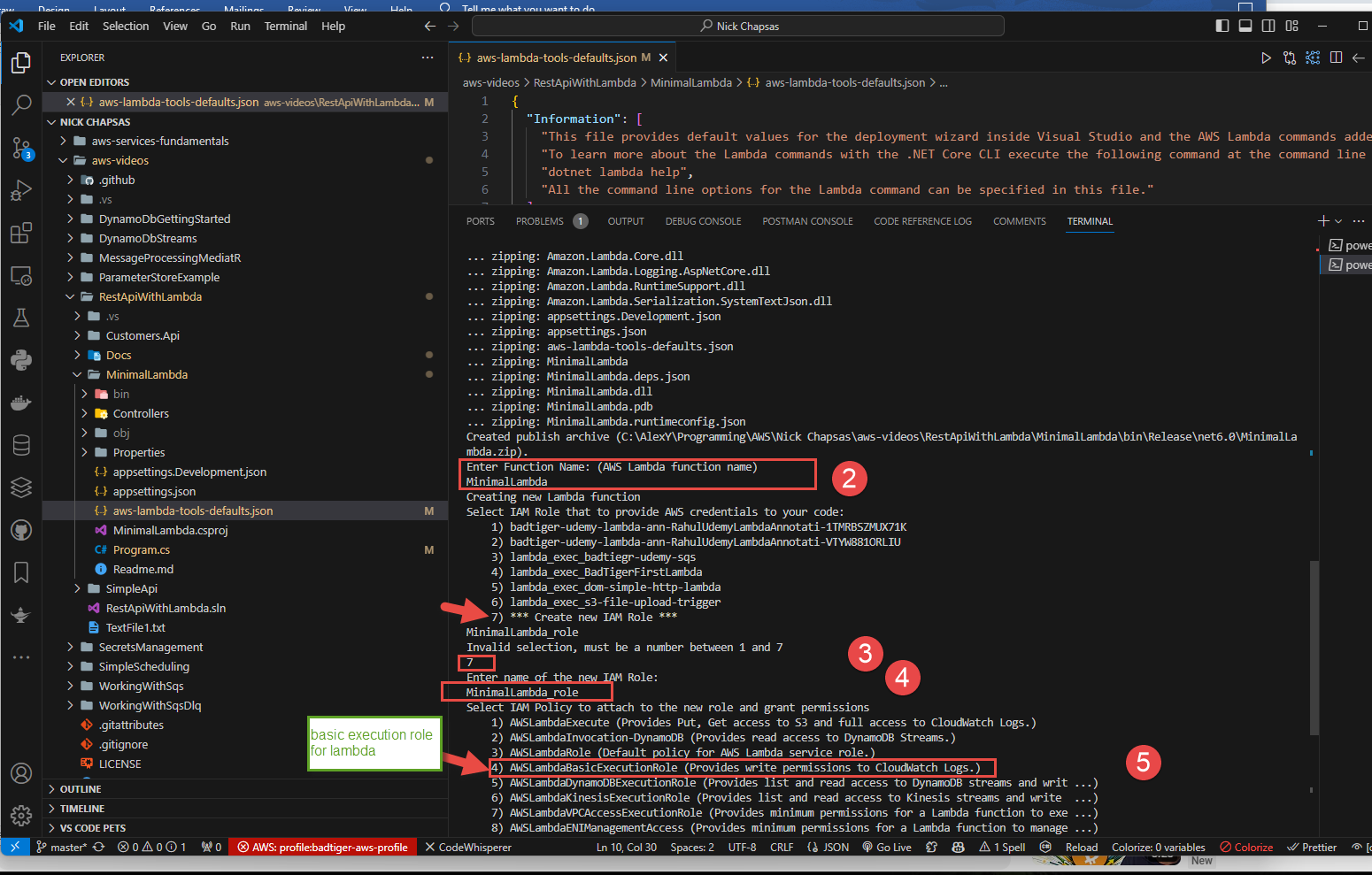
And the project is



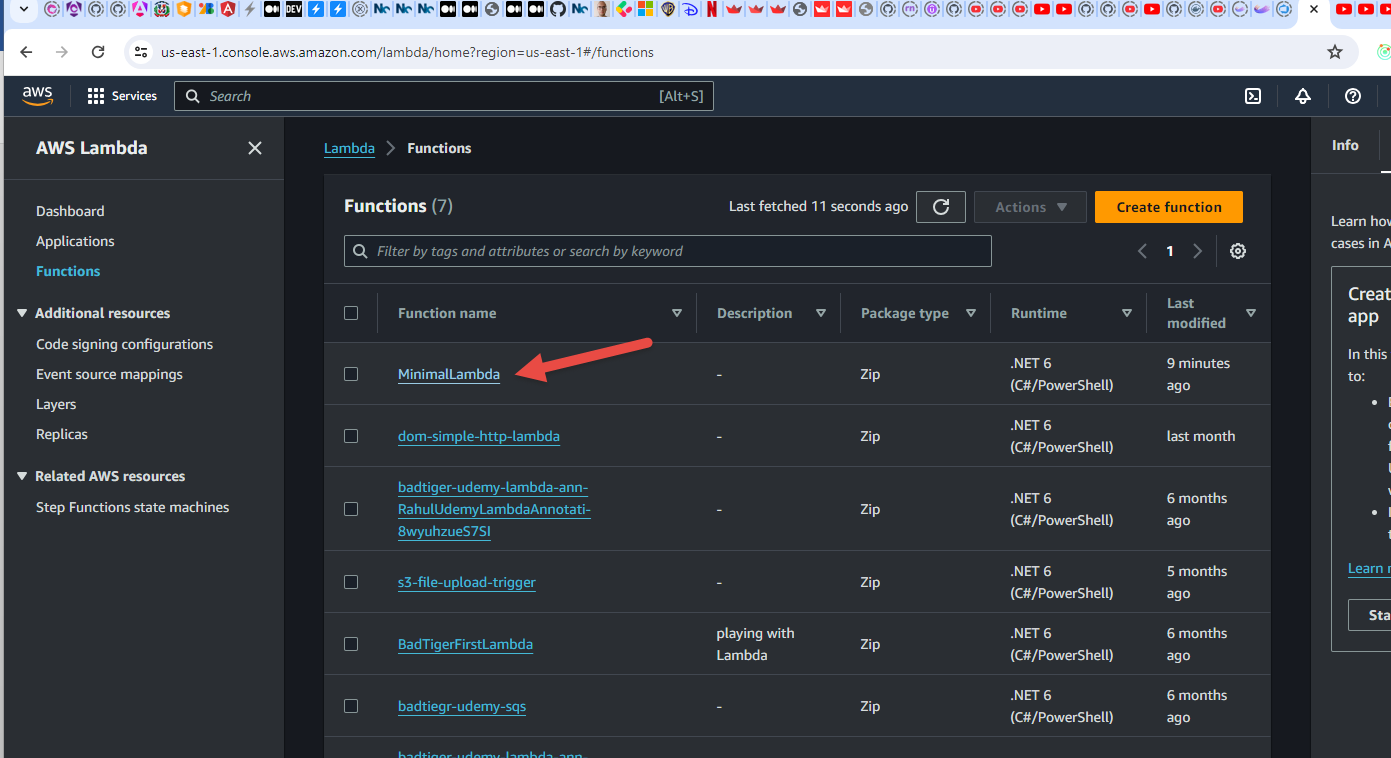


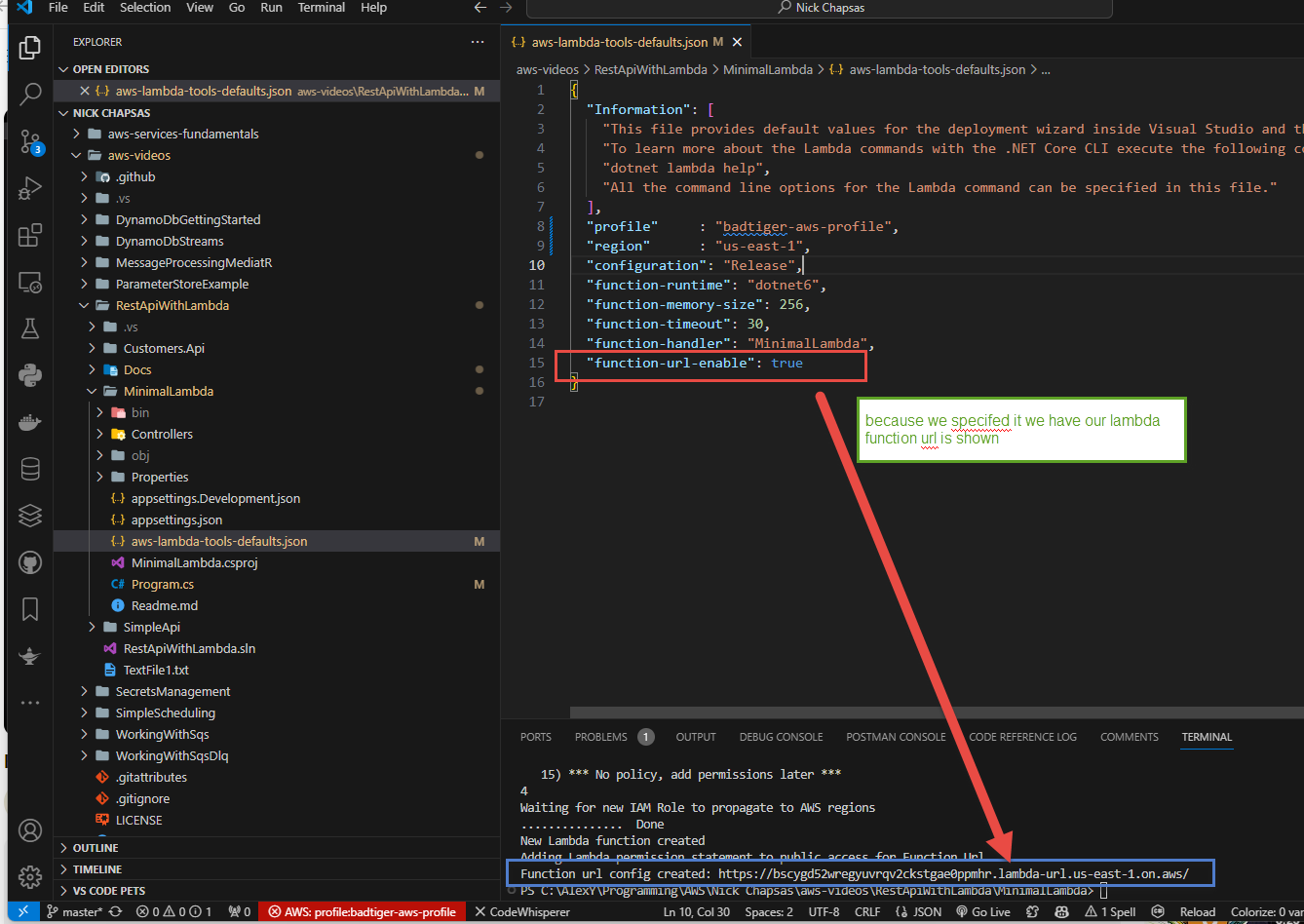
This is a cool stuff is how we deploy our function to AWS:



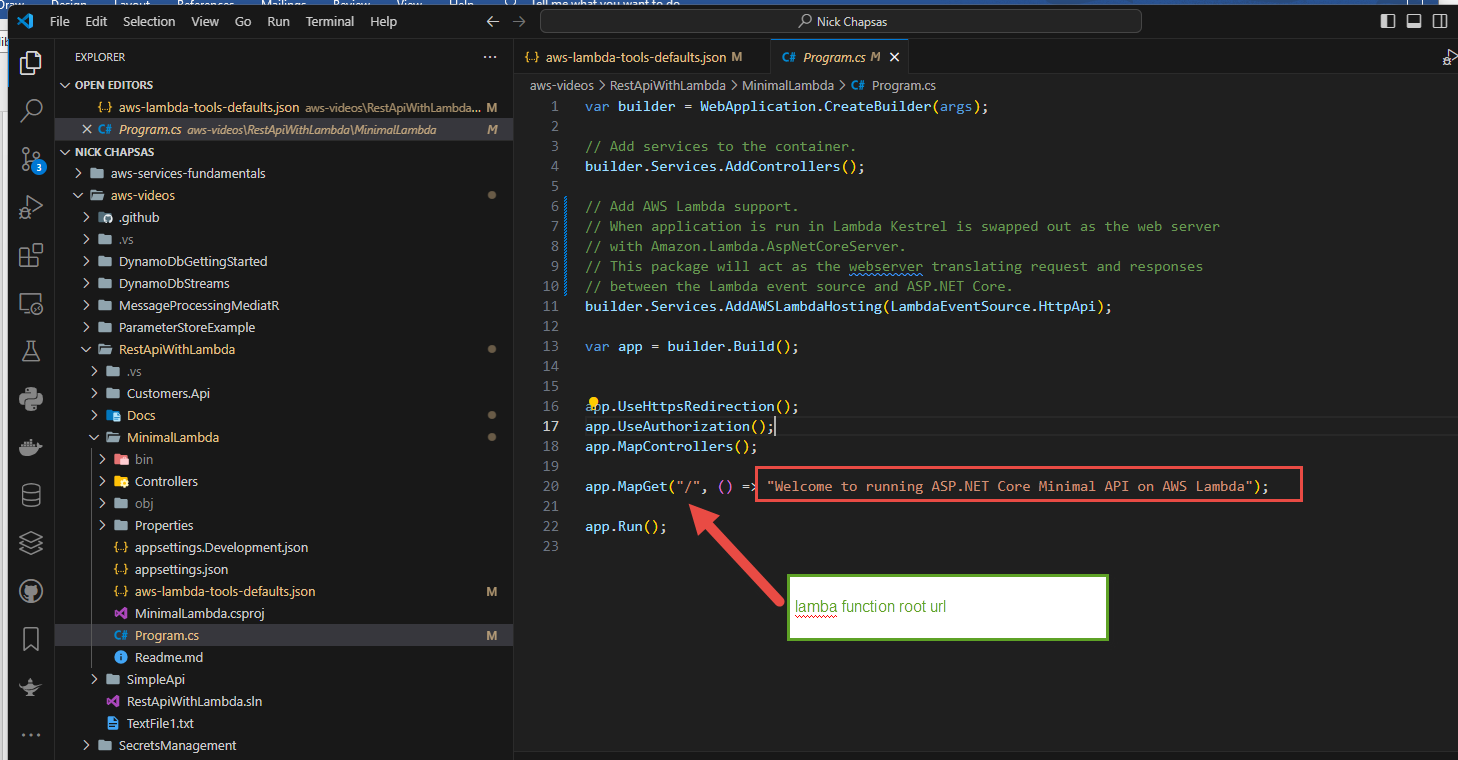


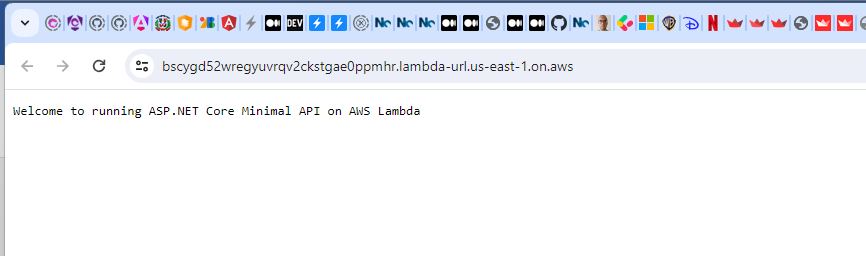
Let’s find it on AWS console



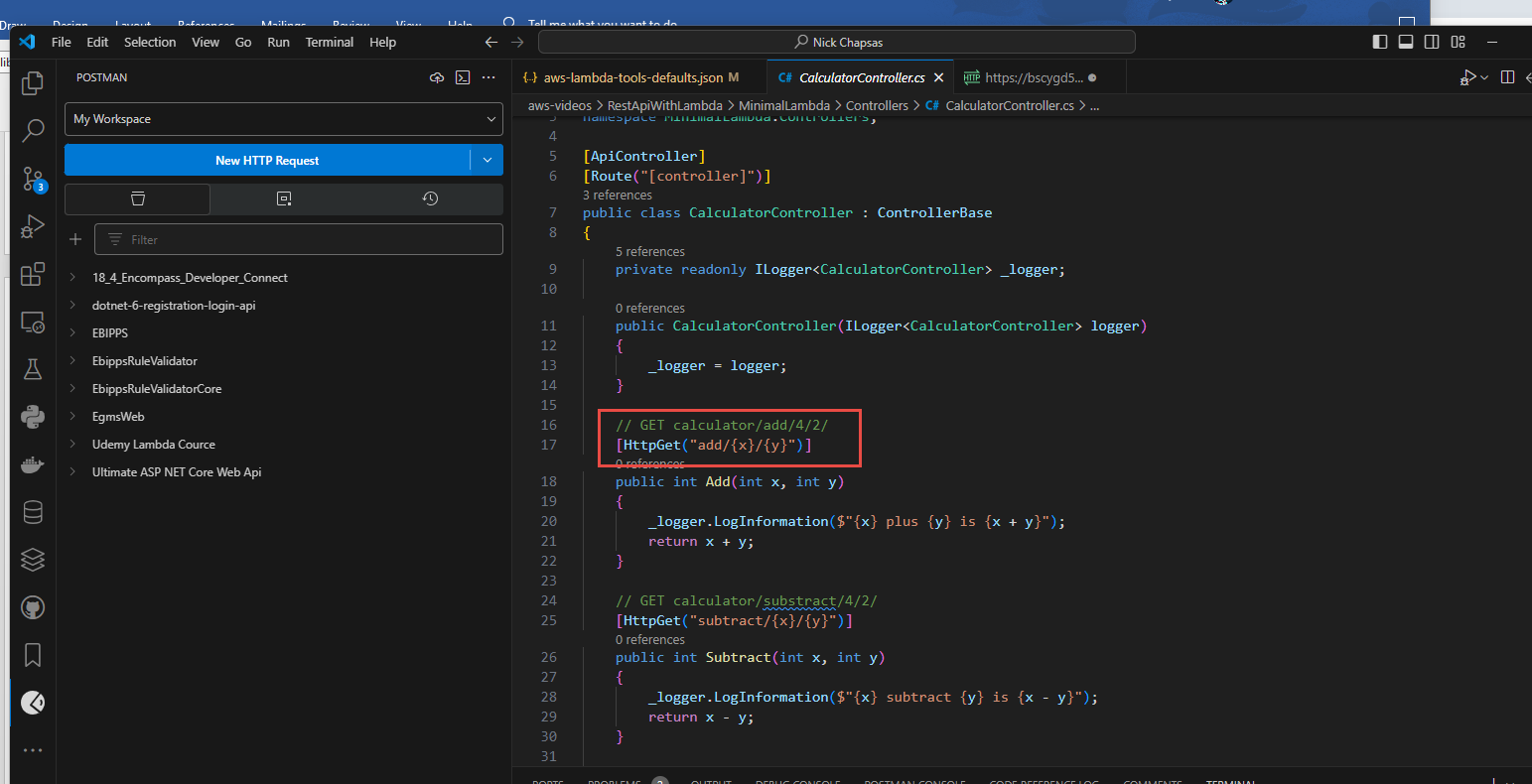


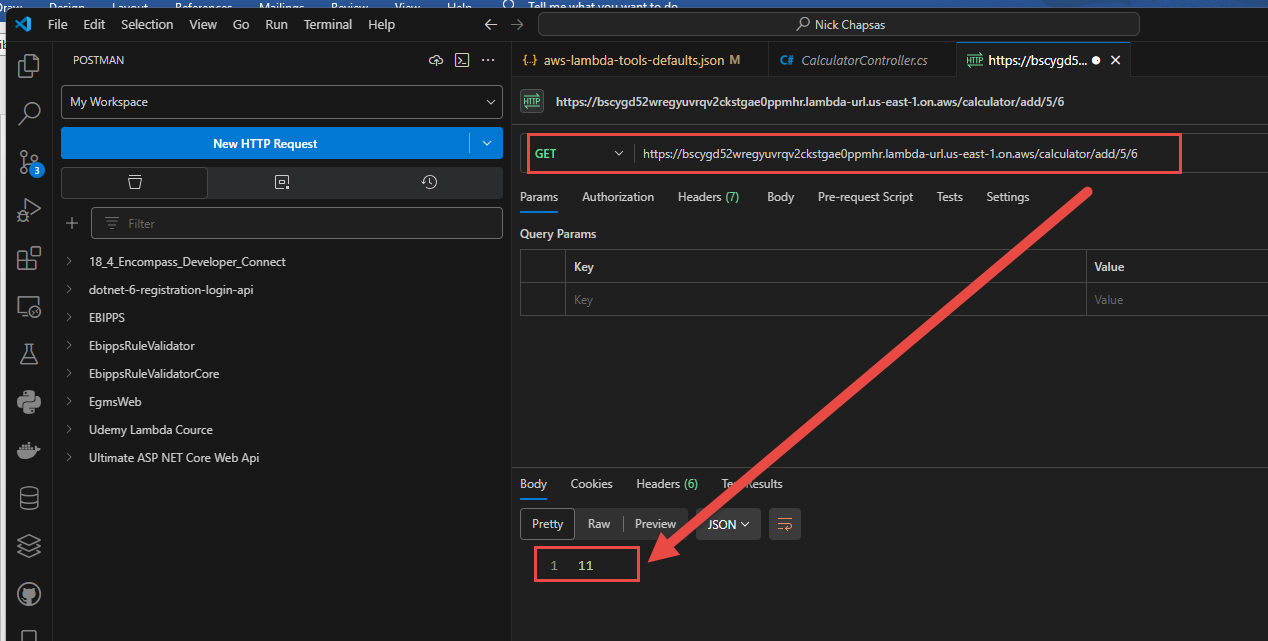
We can click on it and we should see



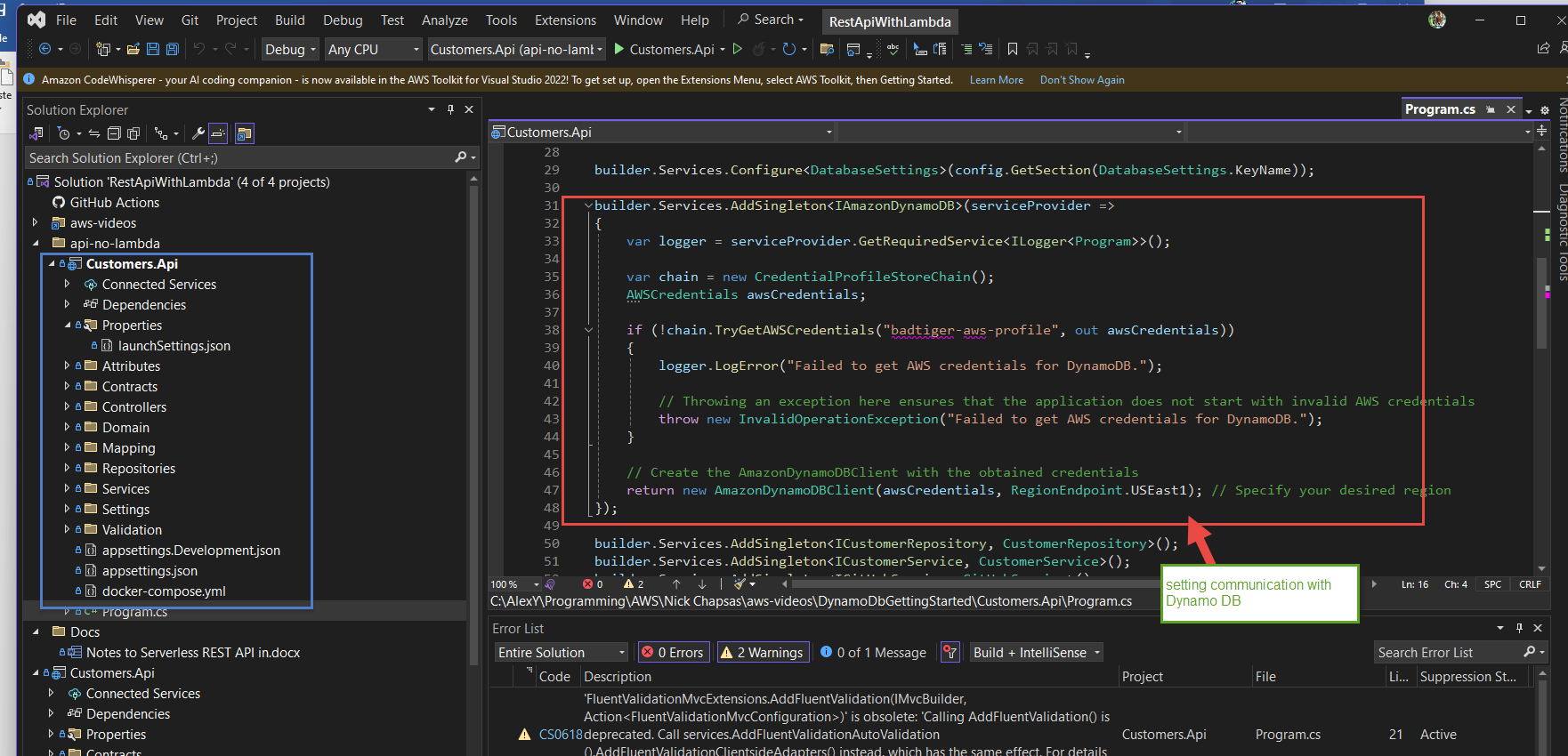


Now we can execute our controllers

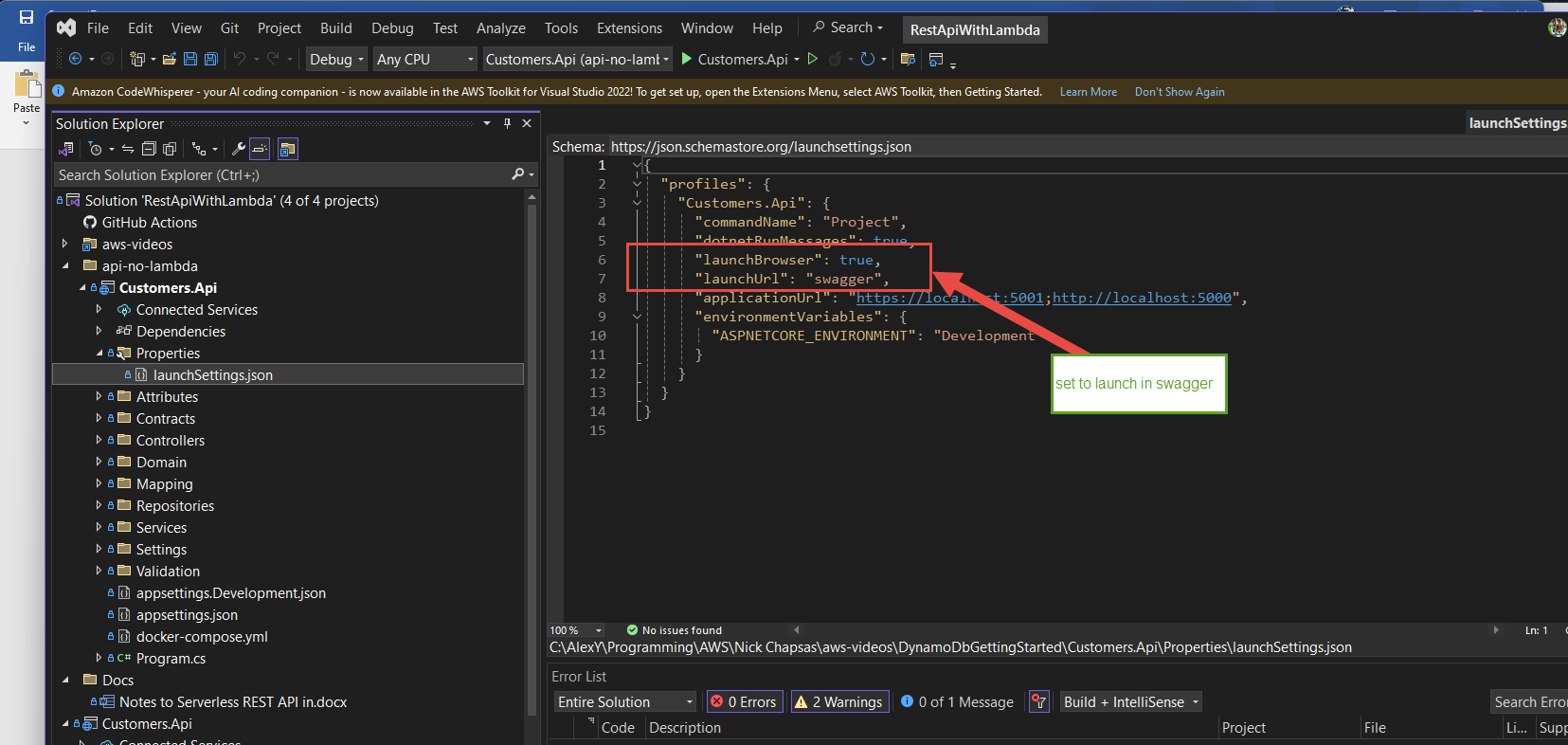


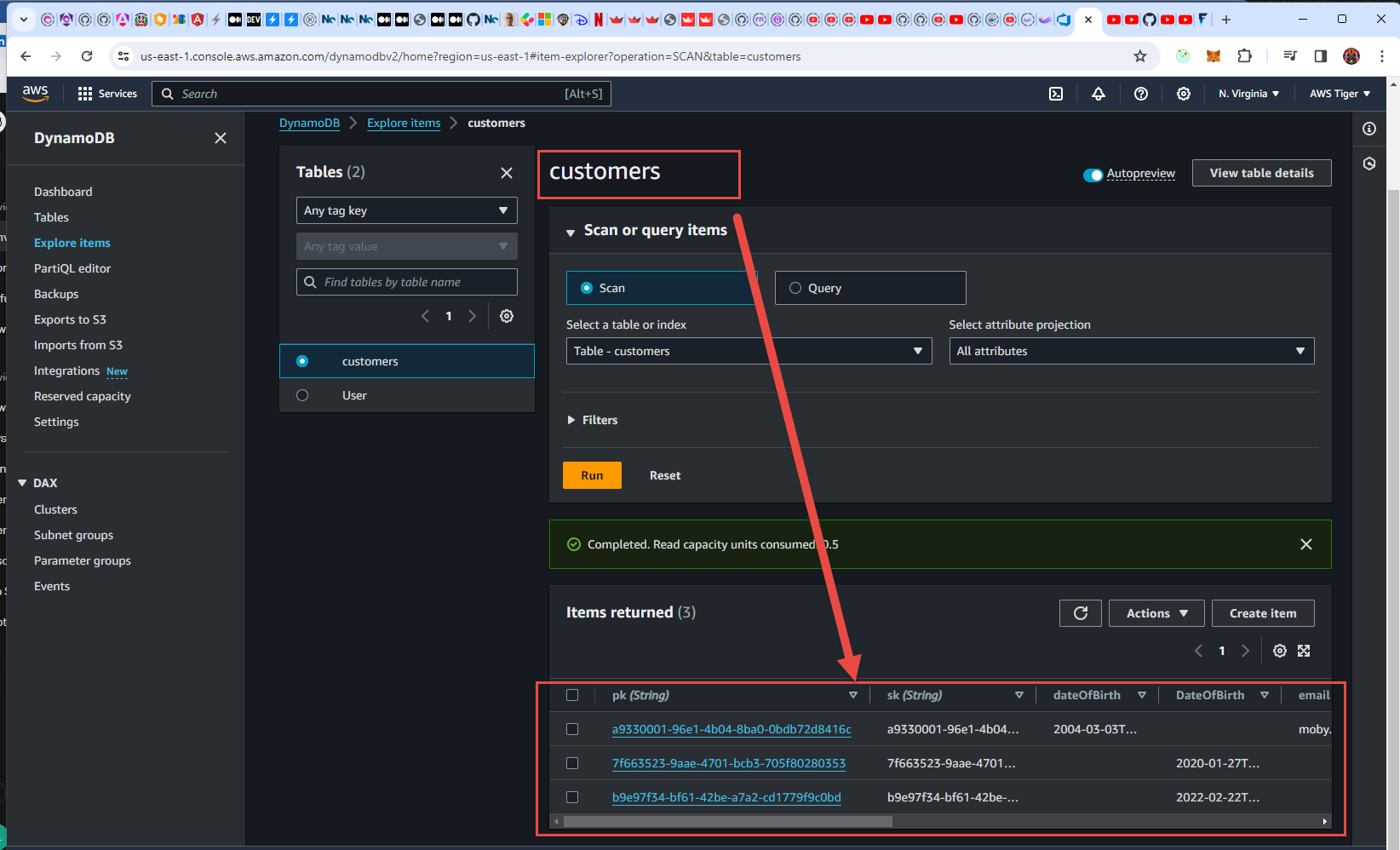


Next let us take a look on the project that is not a lambda but a ASP.Net Core Web API and has communication with DynamoDb

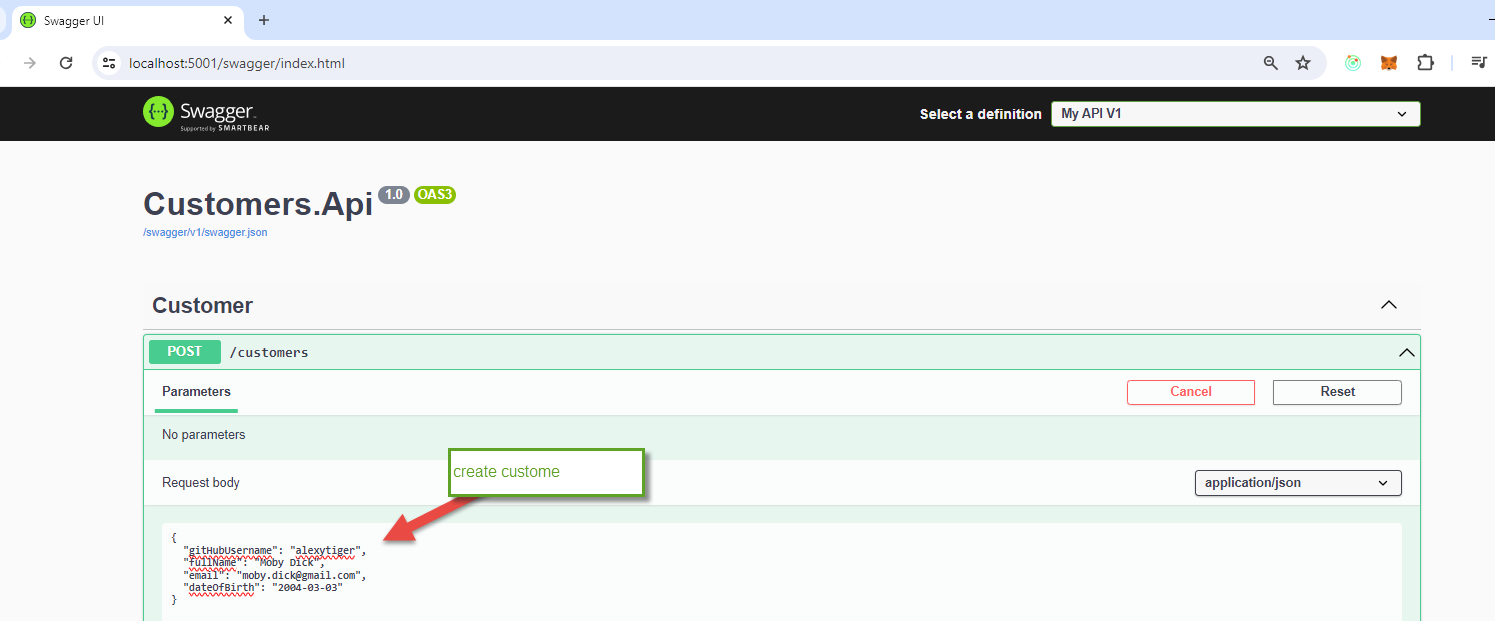


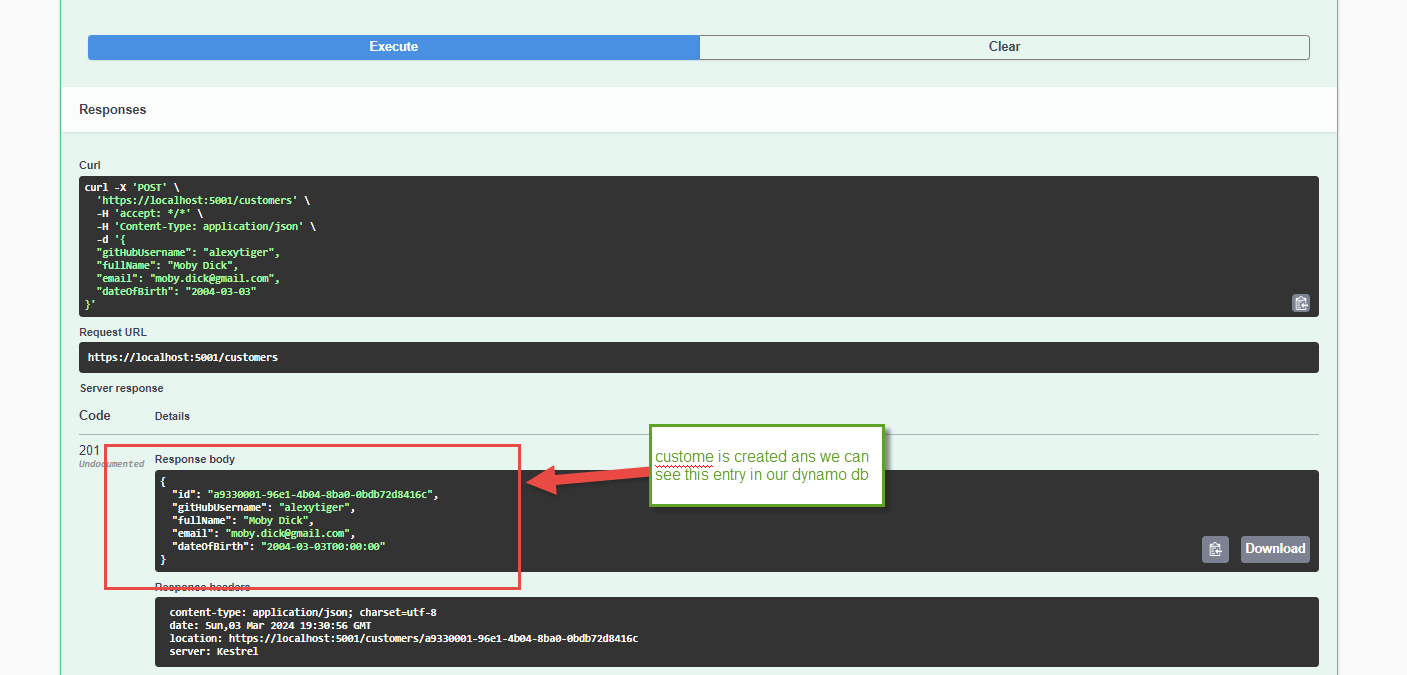


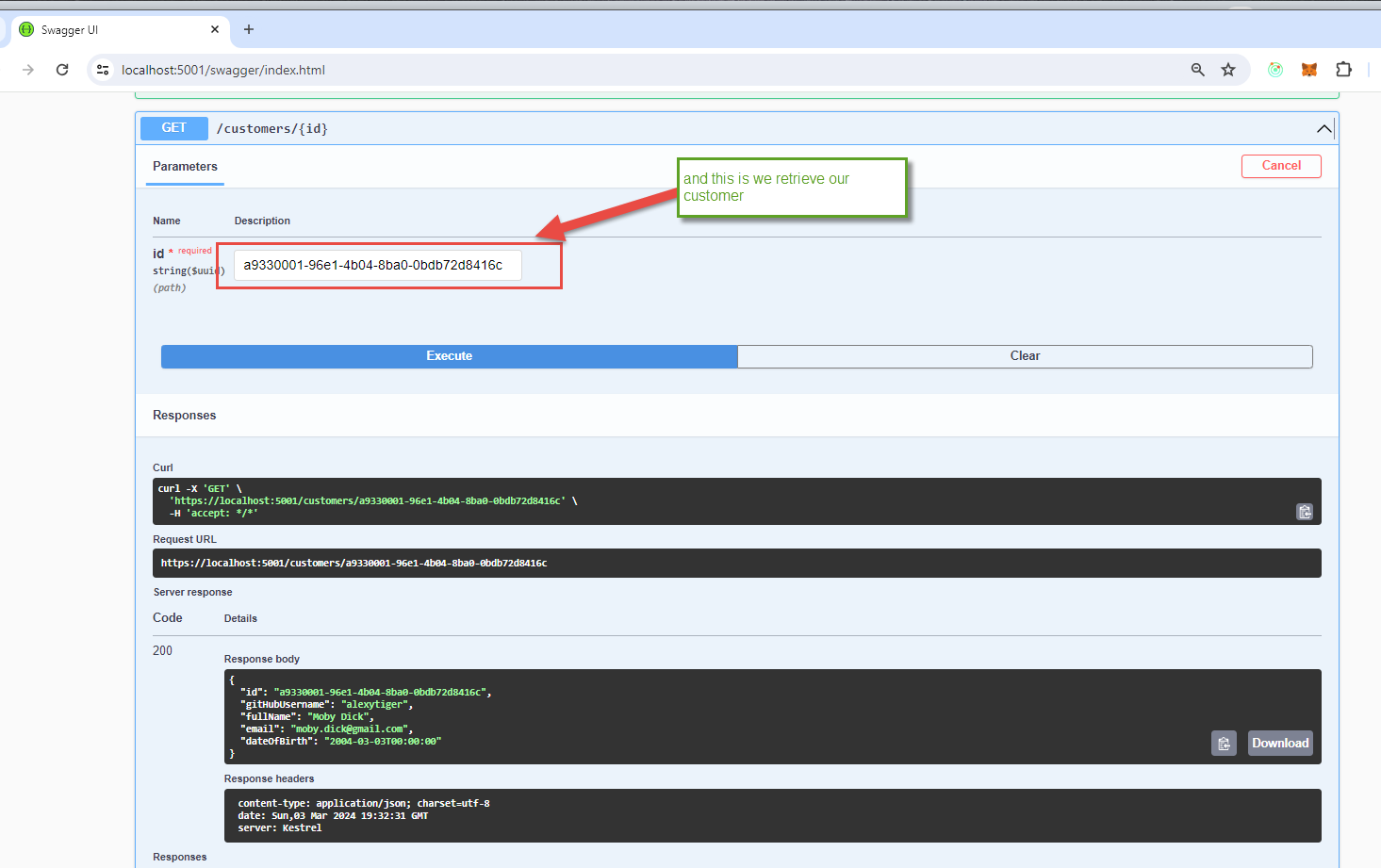


And her eis our dynamp db customer

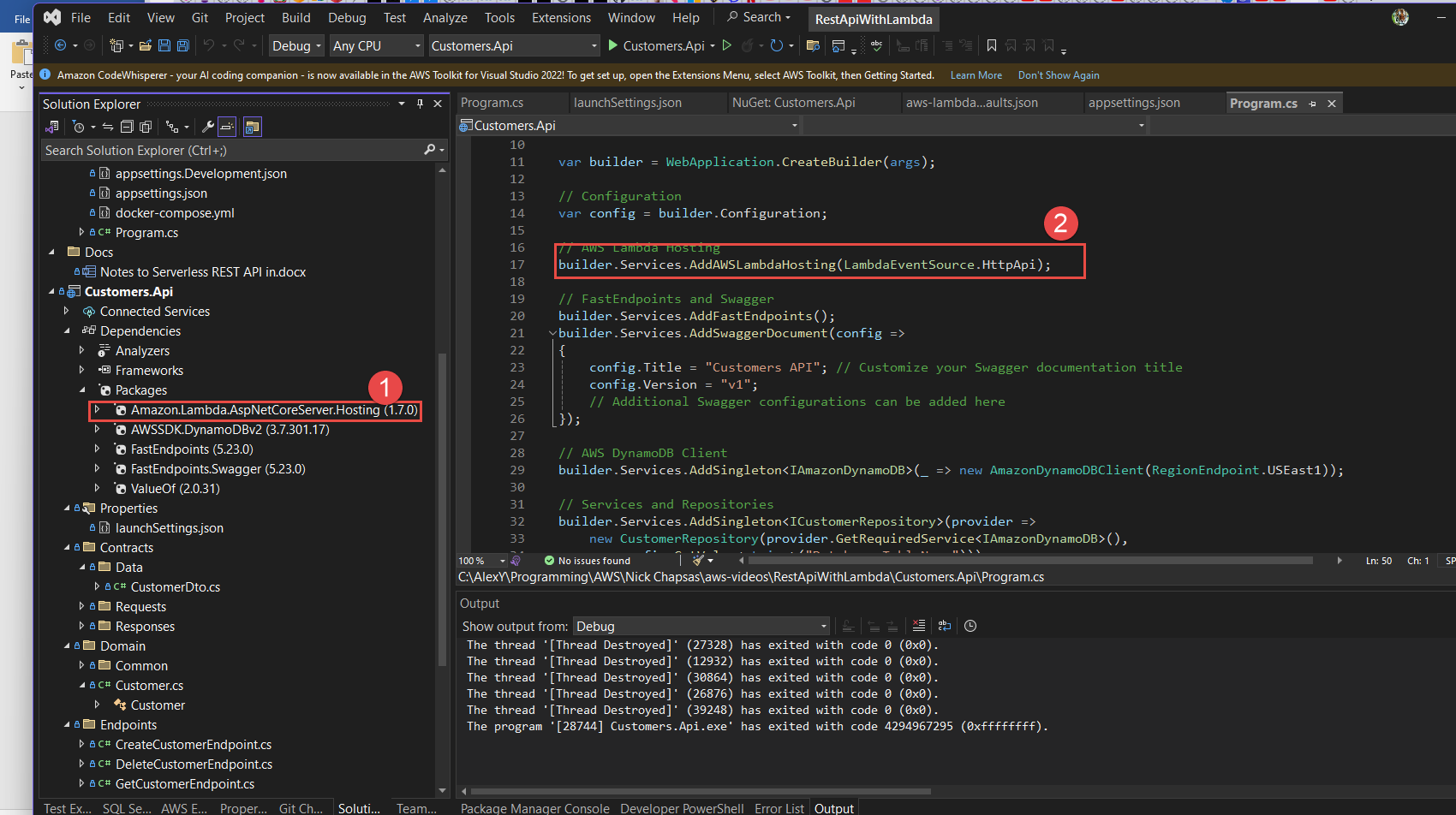
and this is how we create a customer item and how we retrieve it



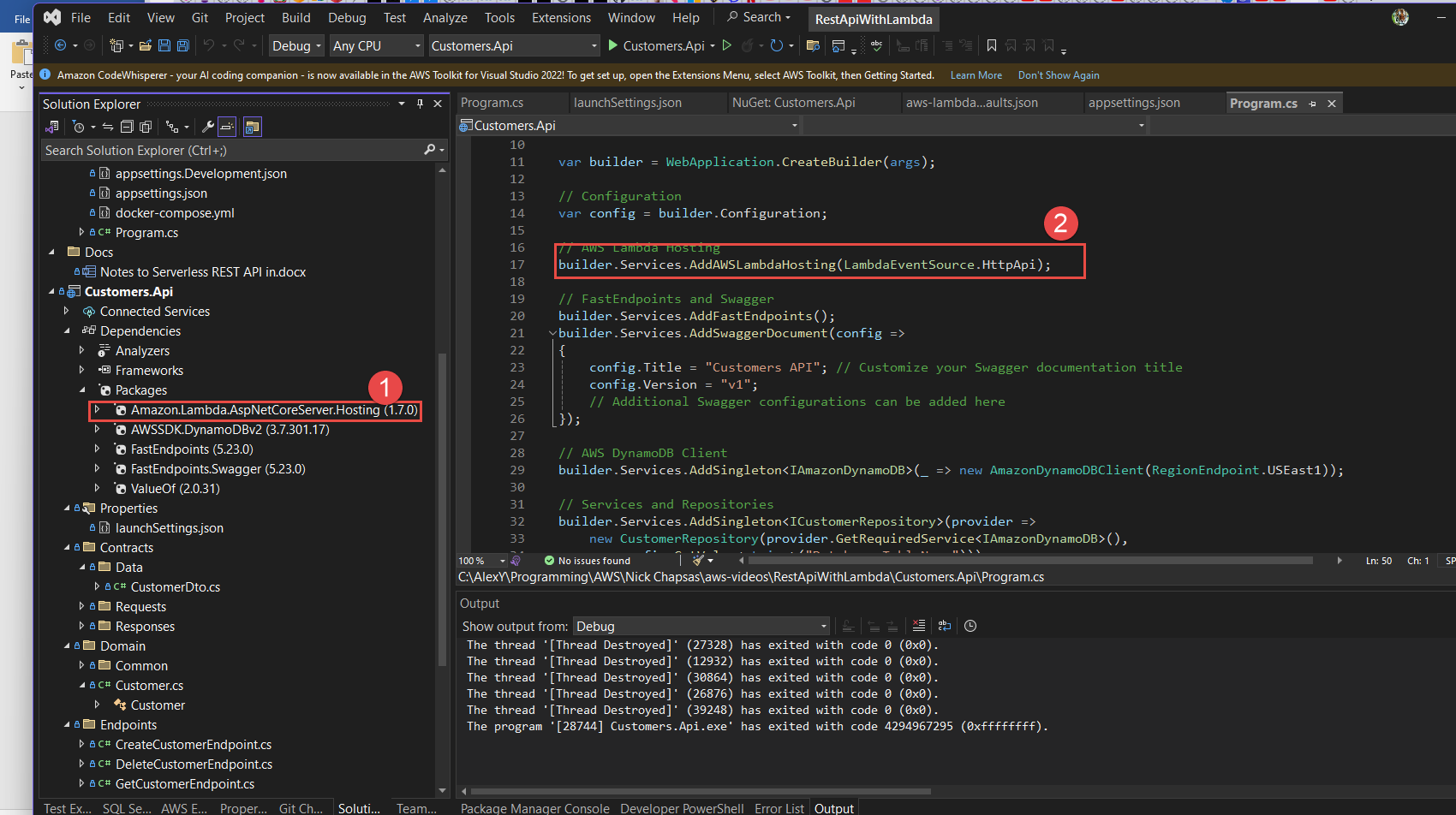




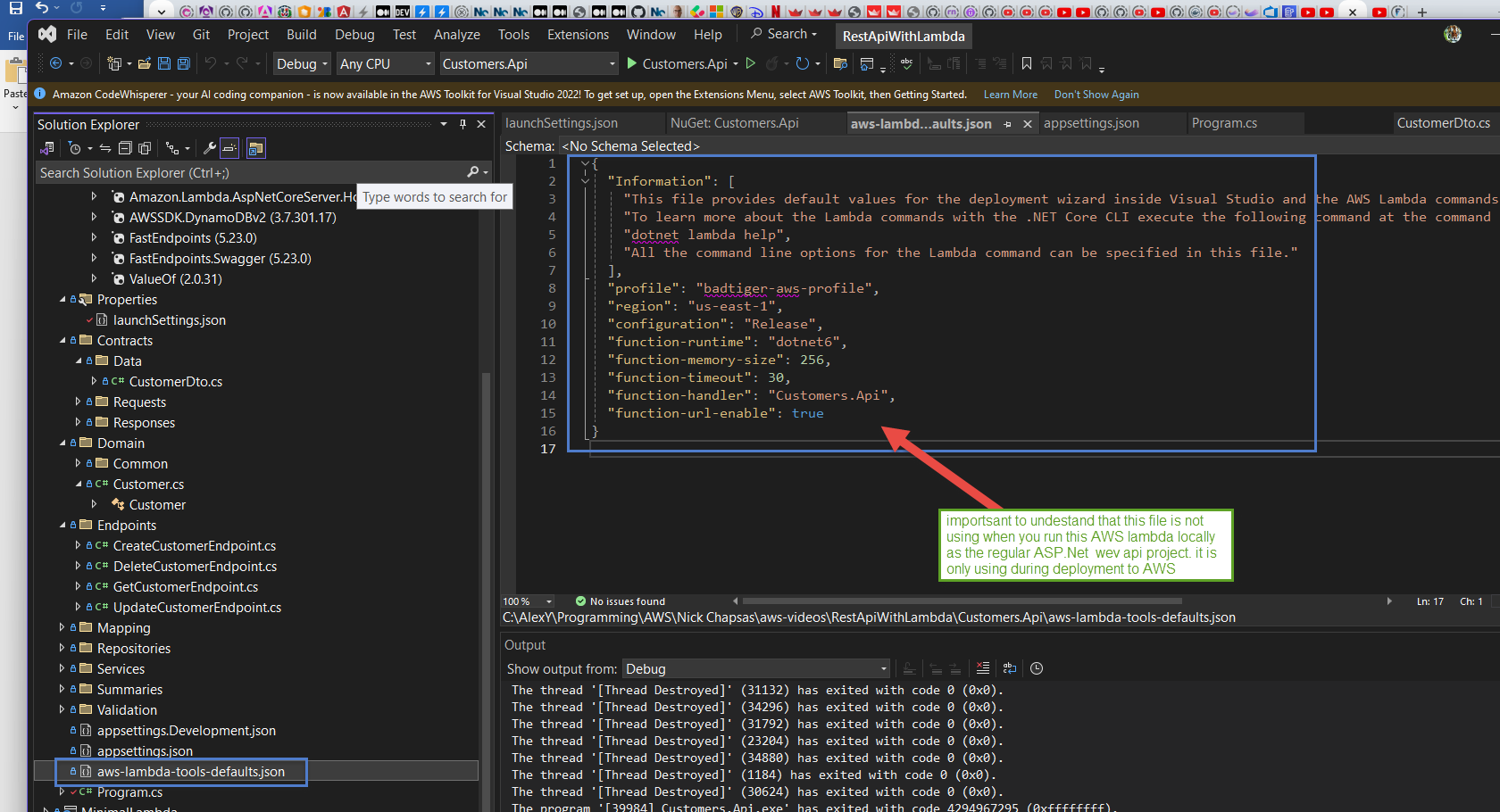
Now , let us take a look on similar project that is actually a lambda one.



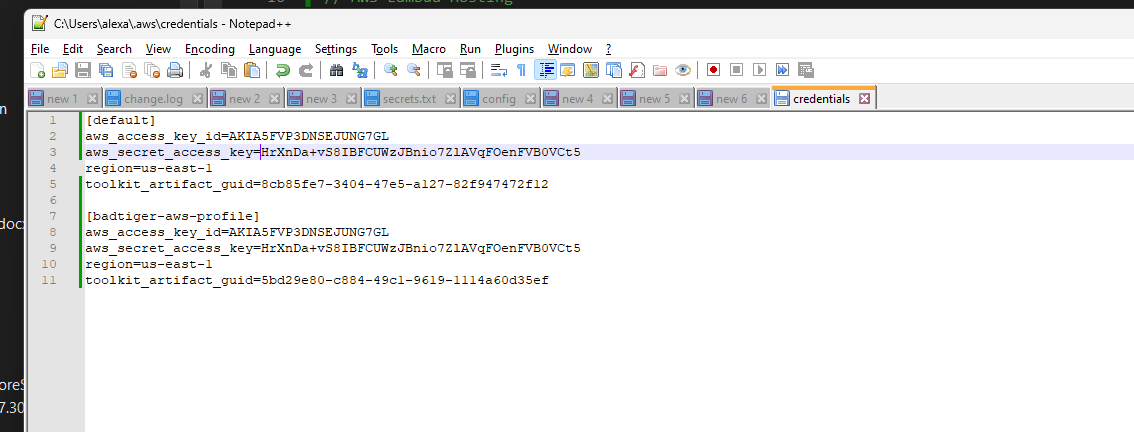
All we needed to do to make it also a s a lambda is this:



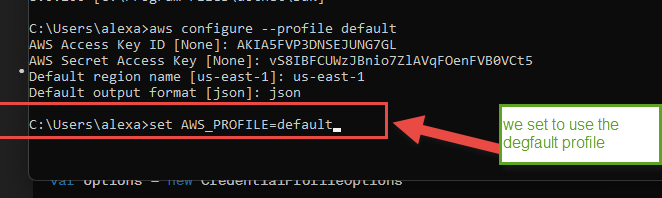
And this:



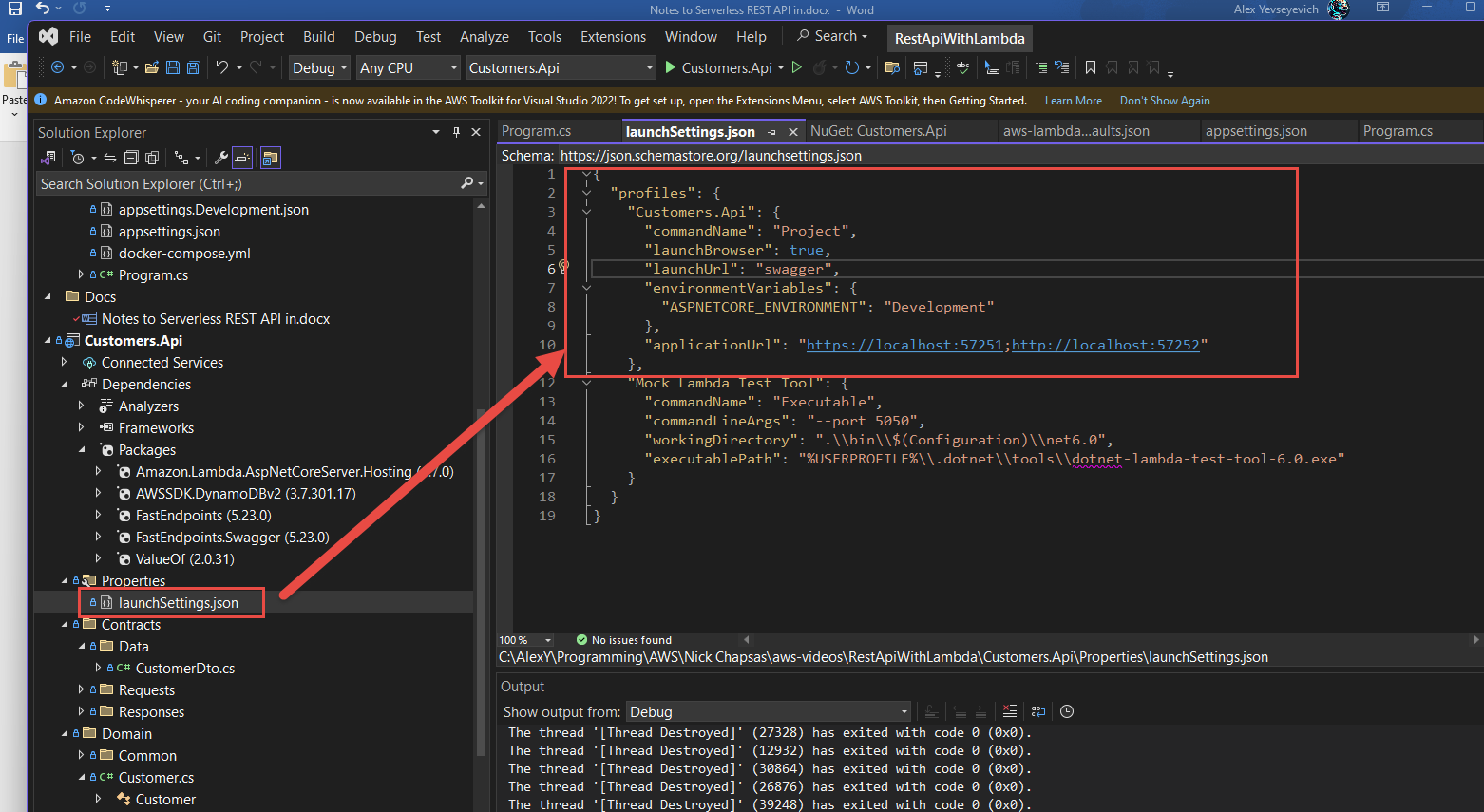
Interesting enough we can still run it locally as the usual ASP.net web api project. Let us do it. Fist let tell our machine which AWS profile to use. I have two:

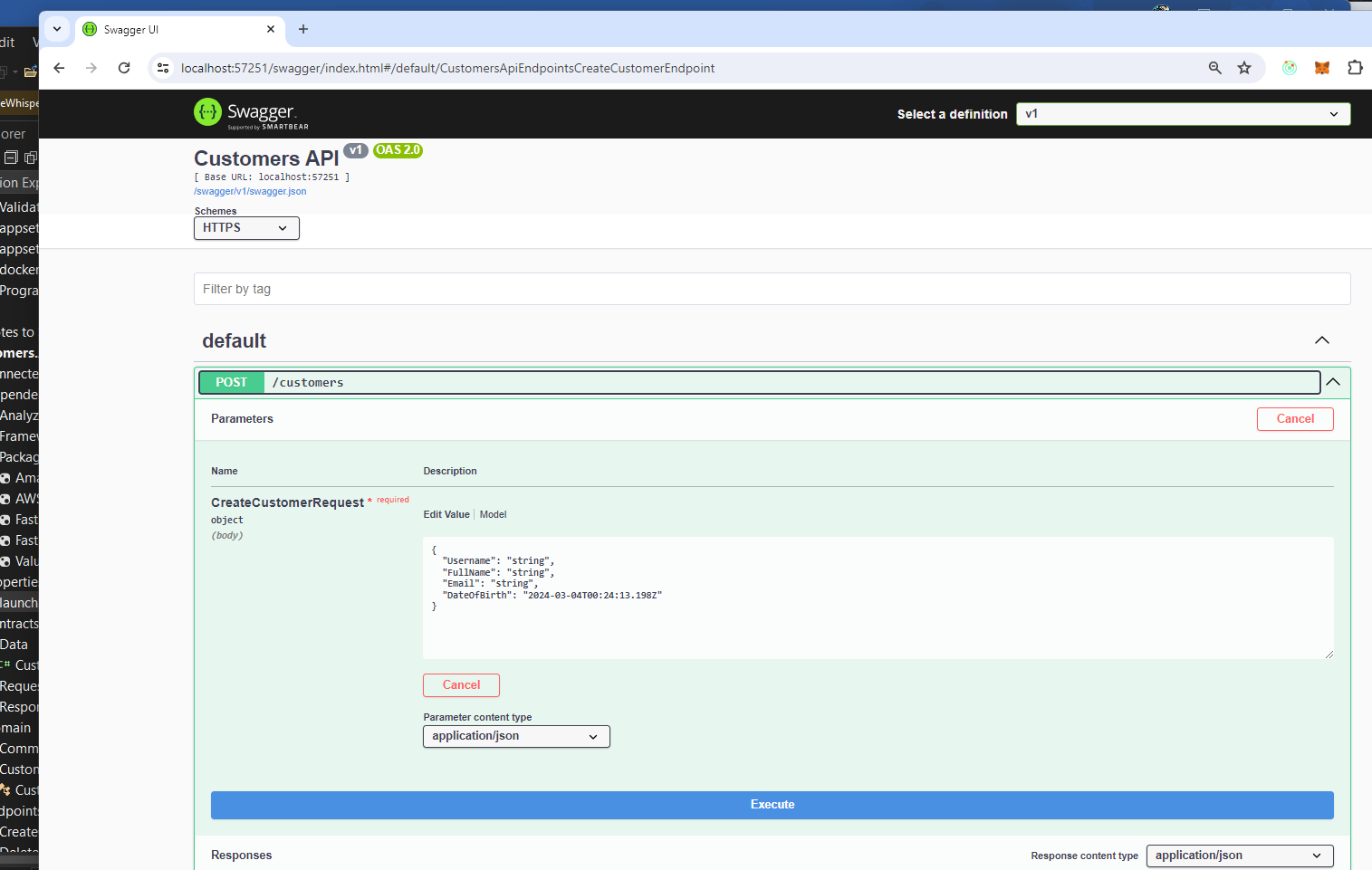


I will tell my machine which one to use because I don’t set it explicitly in the code and let the running environment to pick up it for me as it would do in AWS when I deploy a lambda

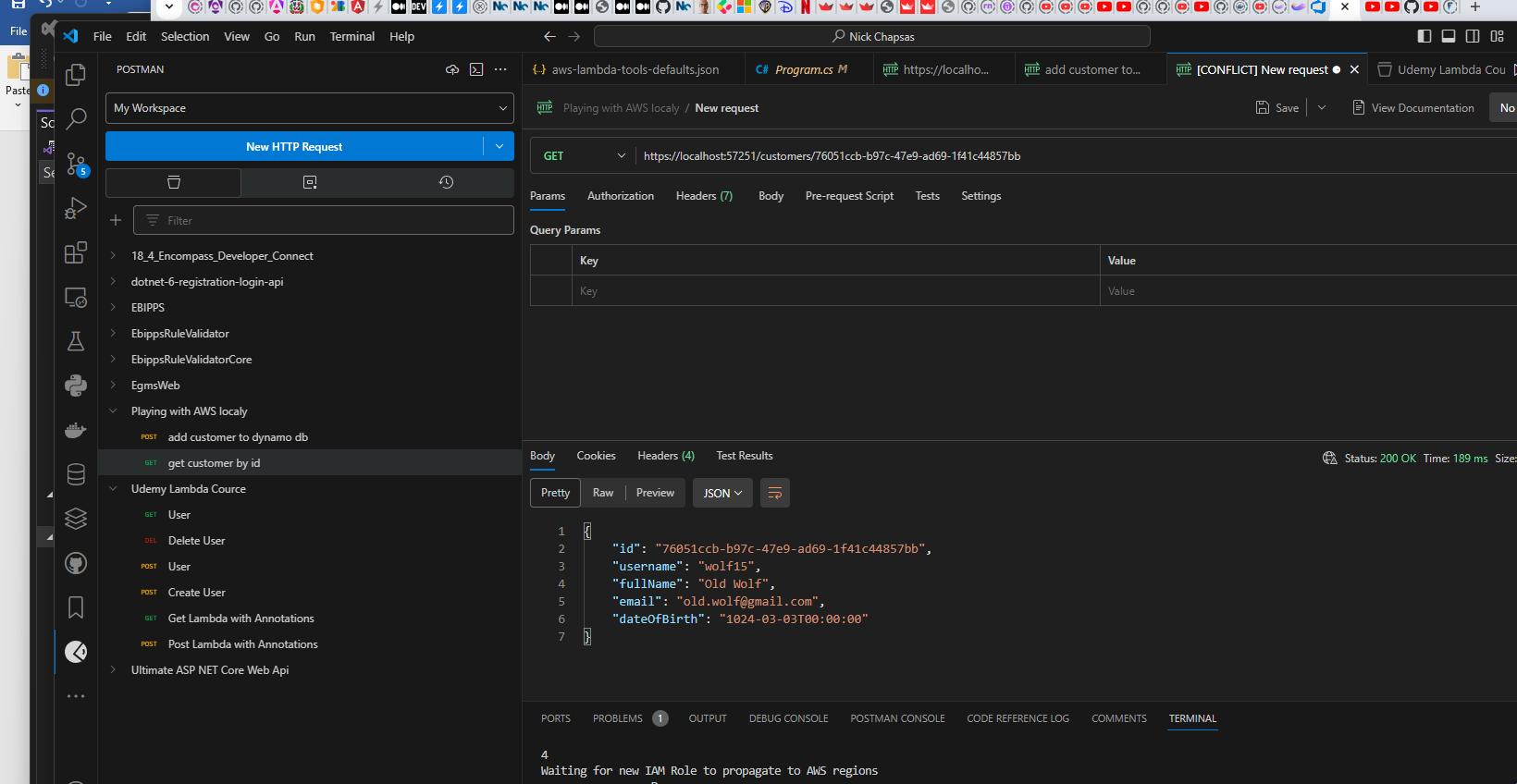


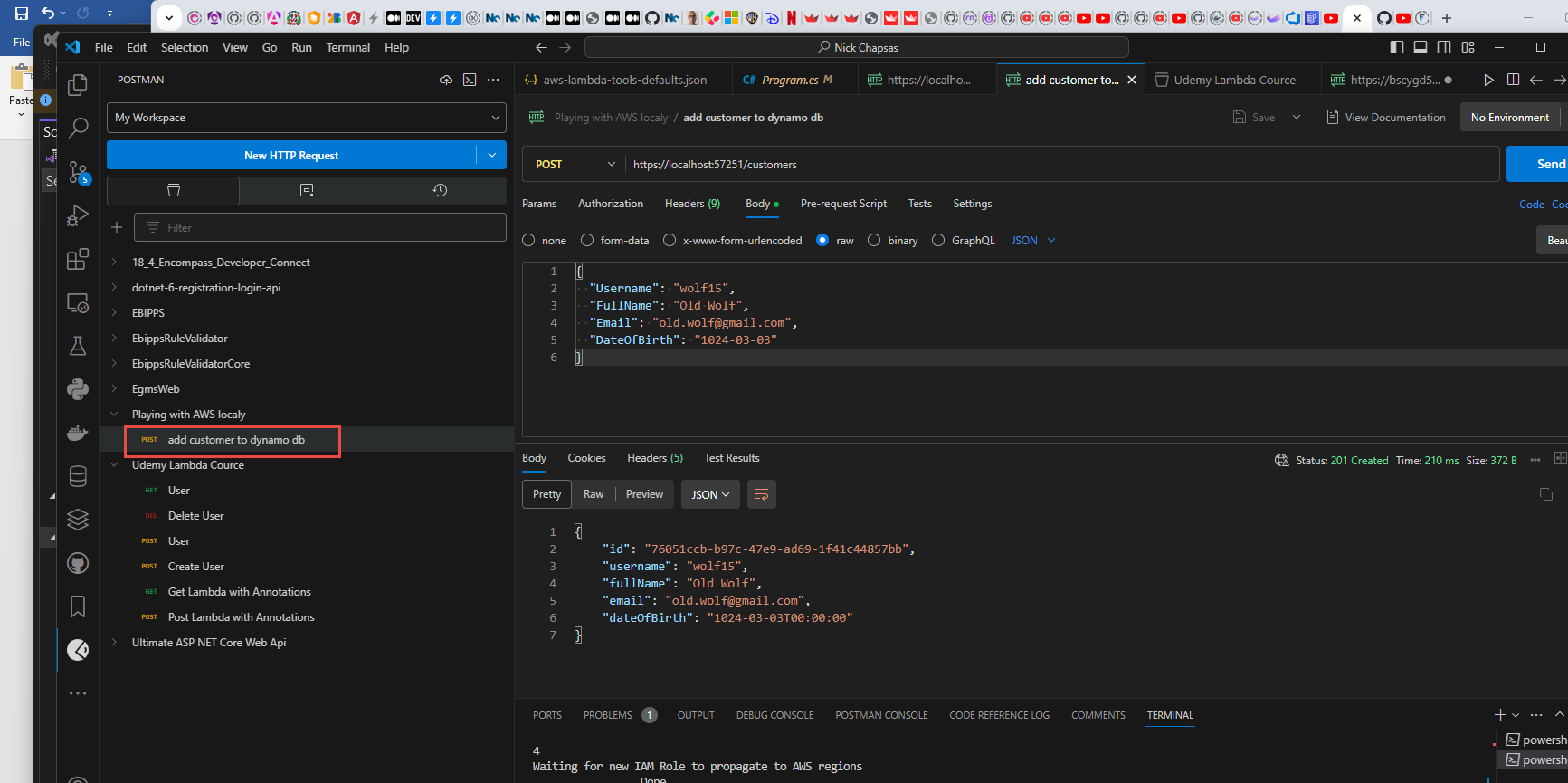
We can also specify that when we run locally we want to launch in swagger

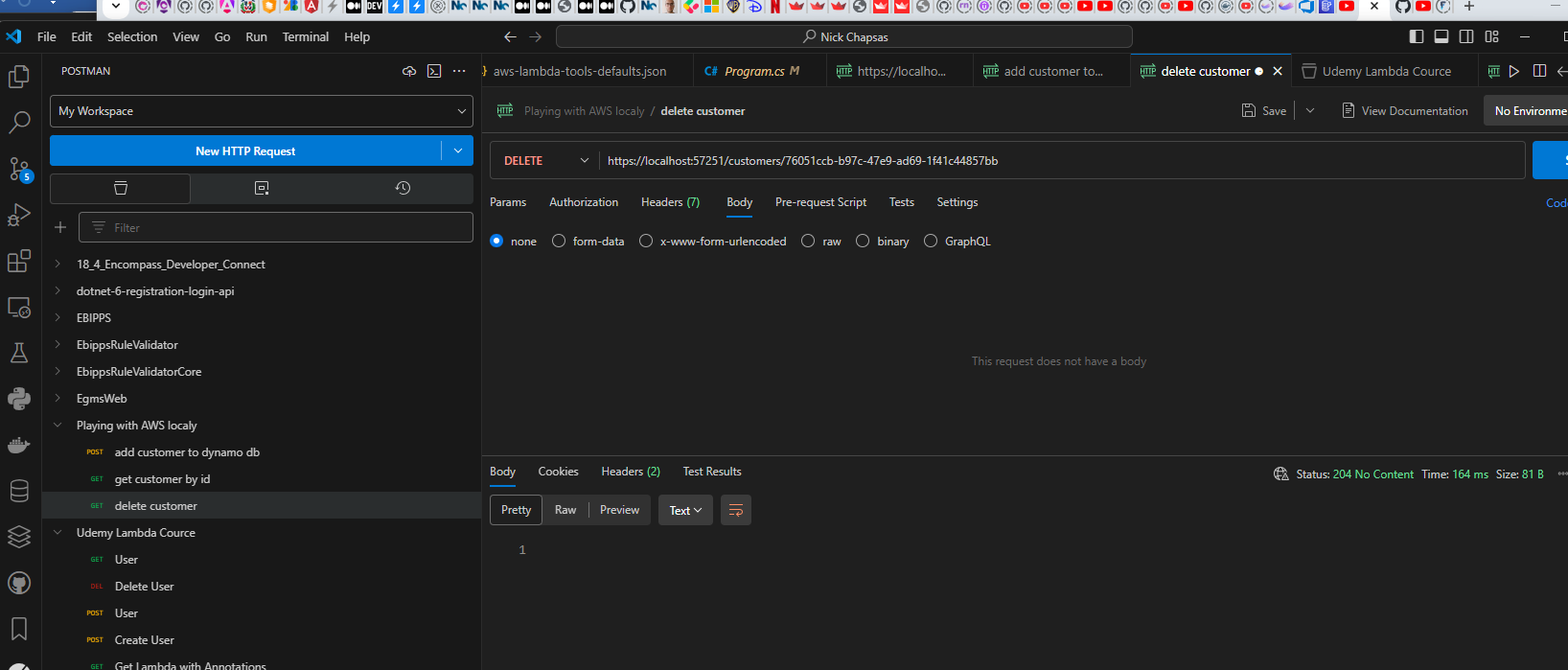




But we are going to use Postman to execute APIs (somehow swagger is no taking properly some api, maybe because I am using this FastEndpoint library here)

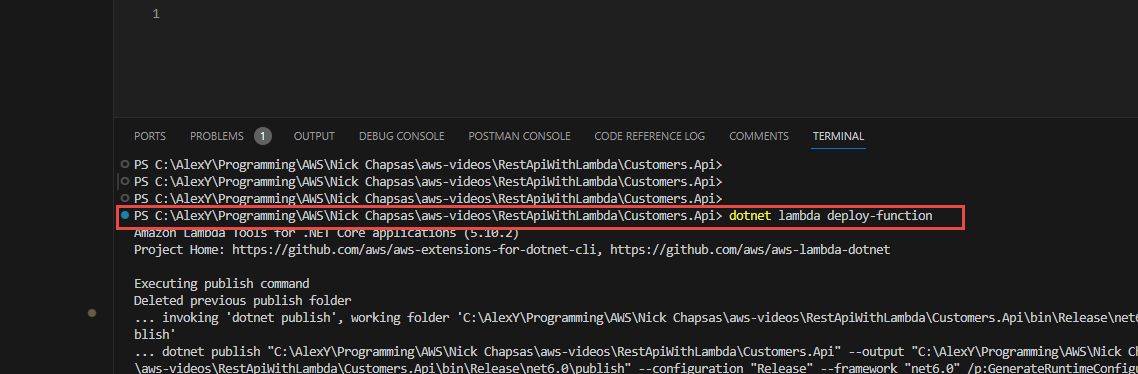


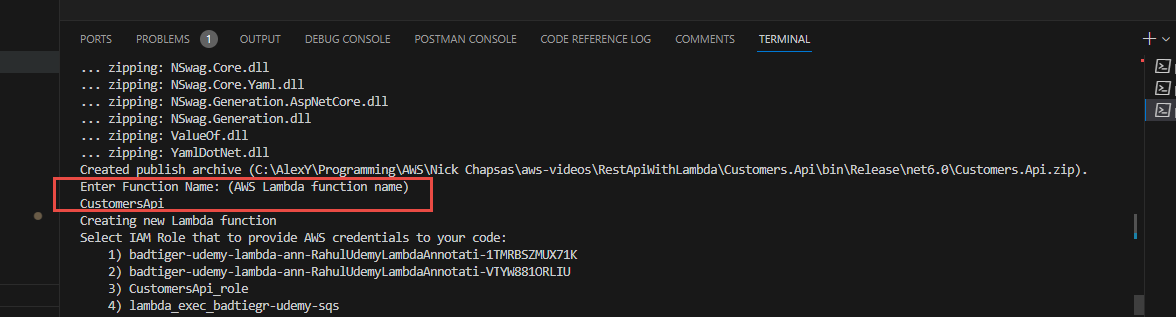


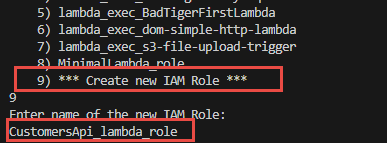


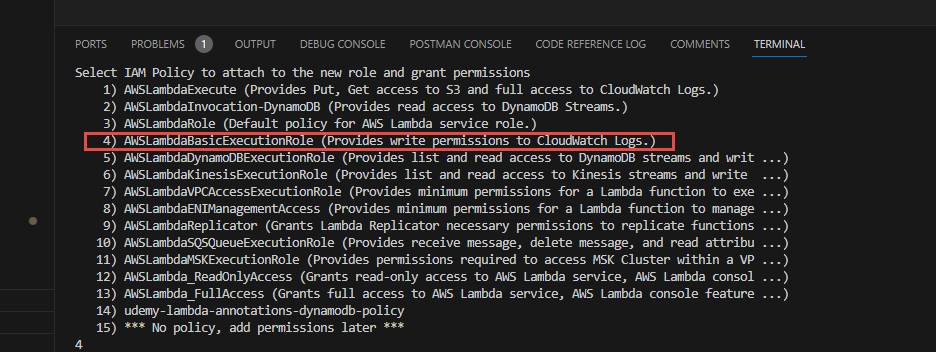
And we can validate that these item have been create in our customers dynamo database

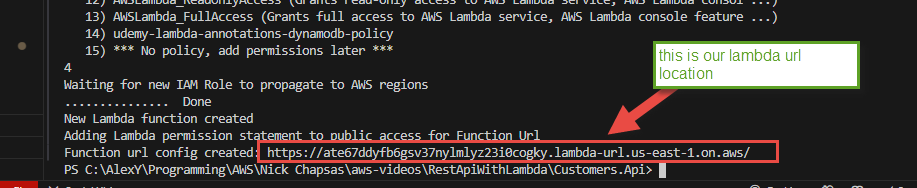
Now , let us deploy our lambda



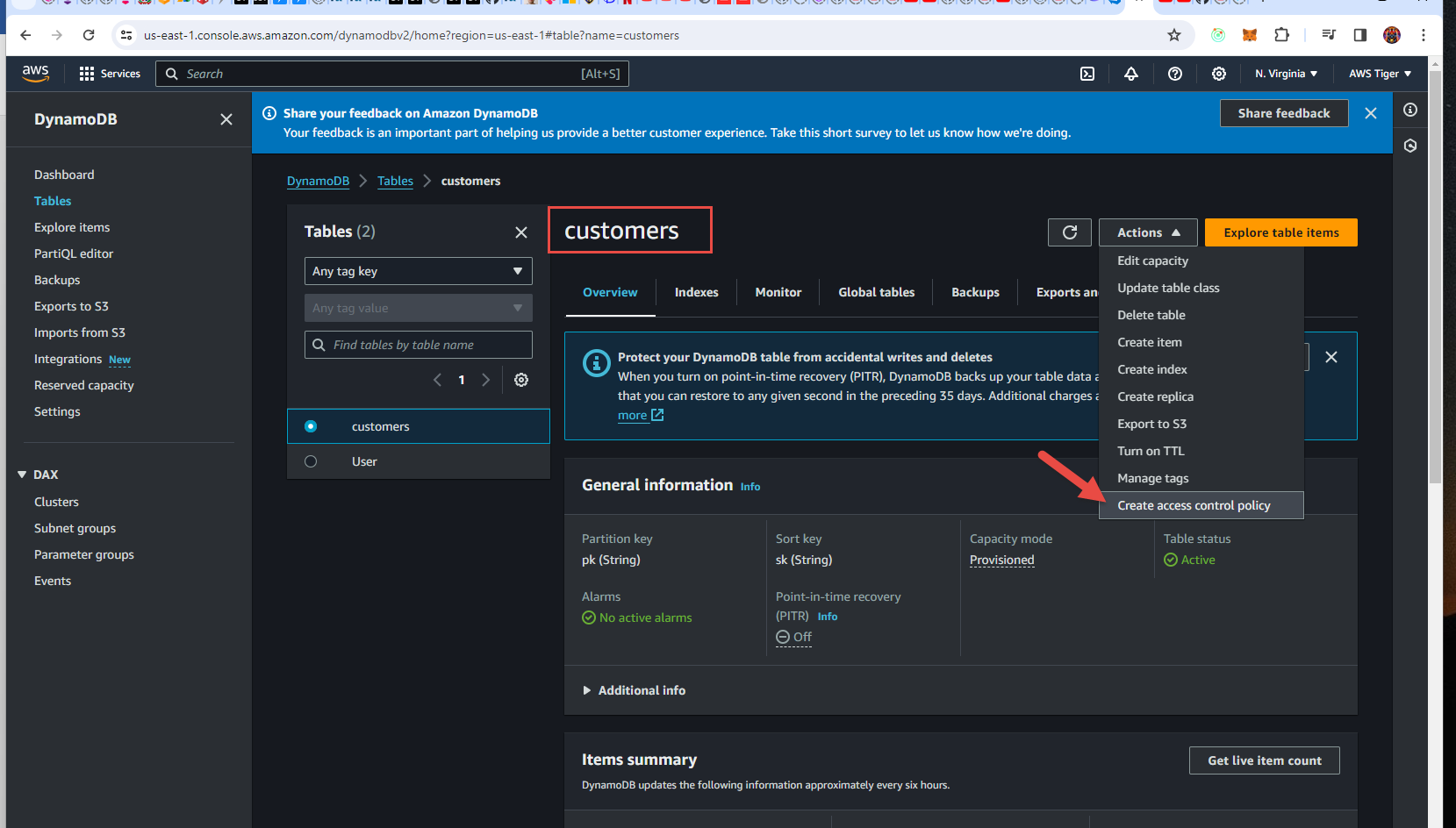


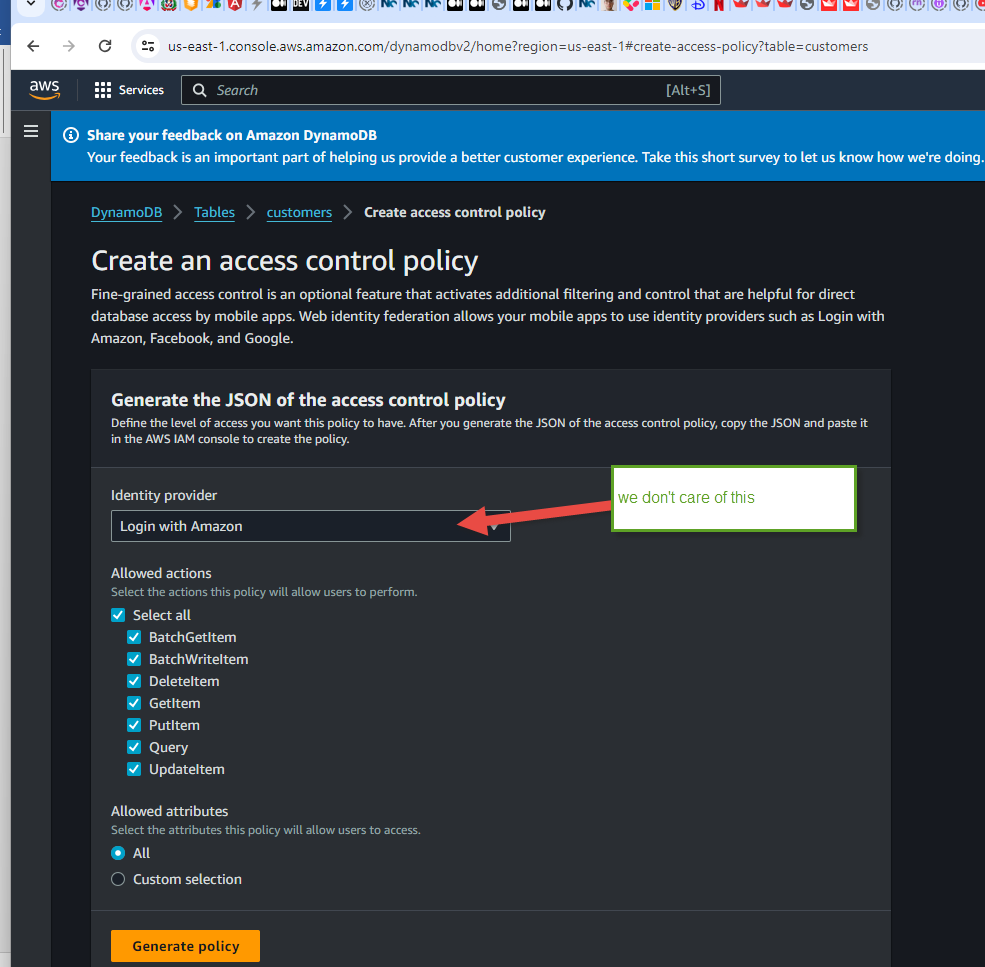


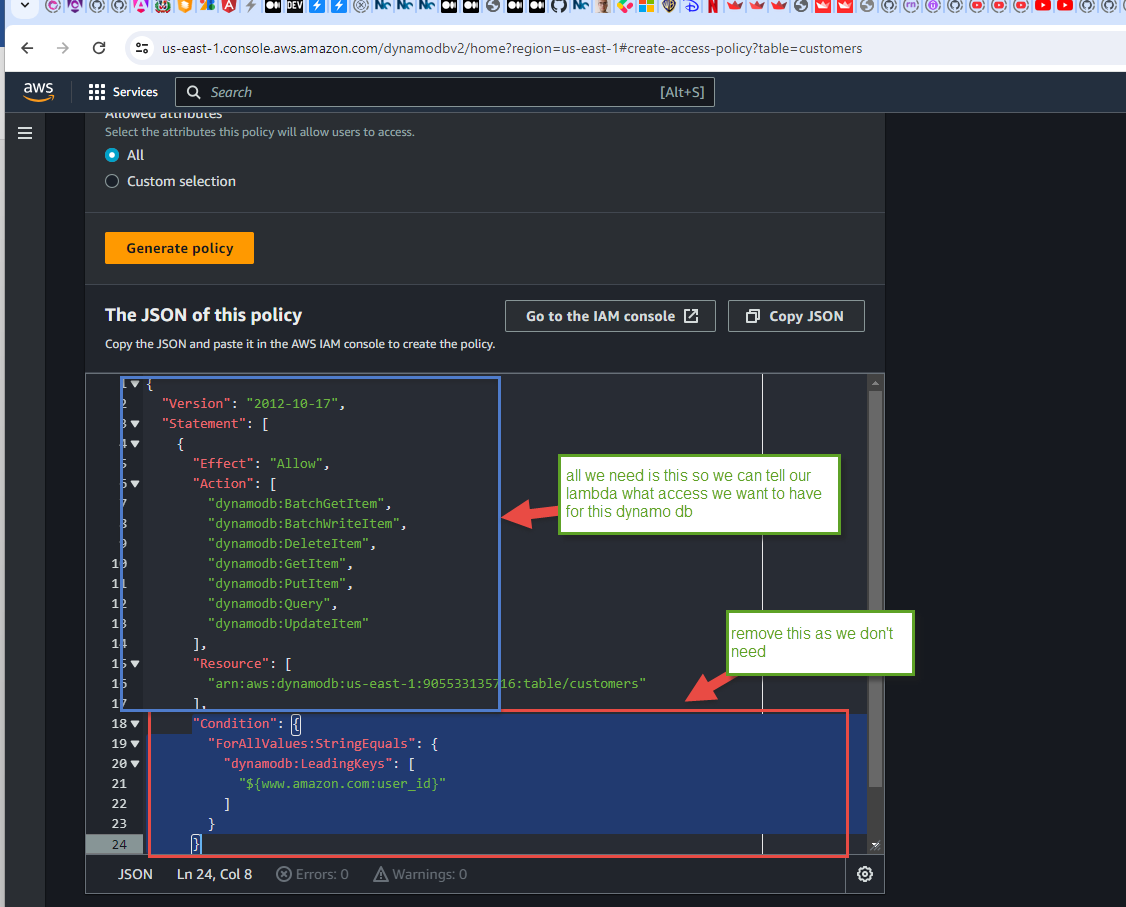


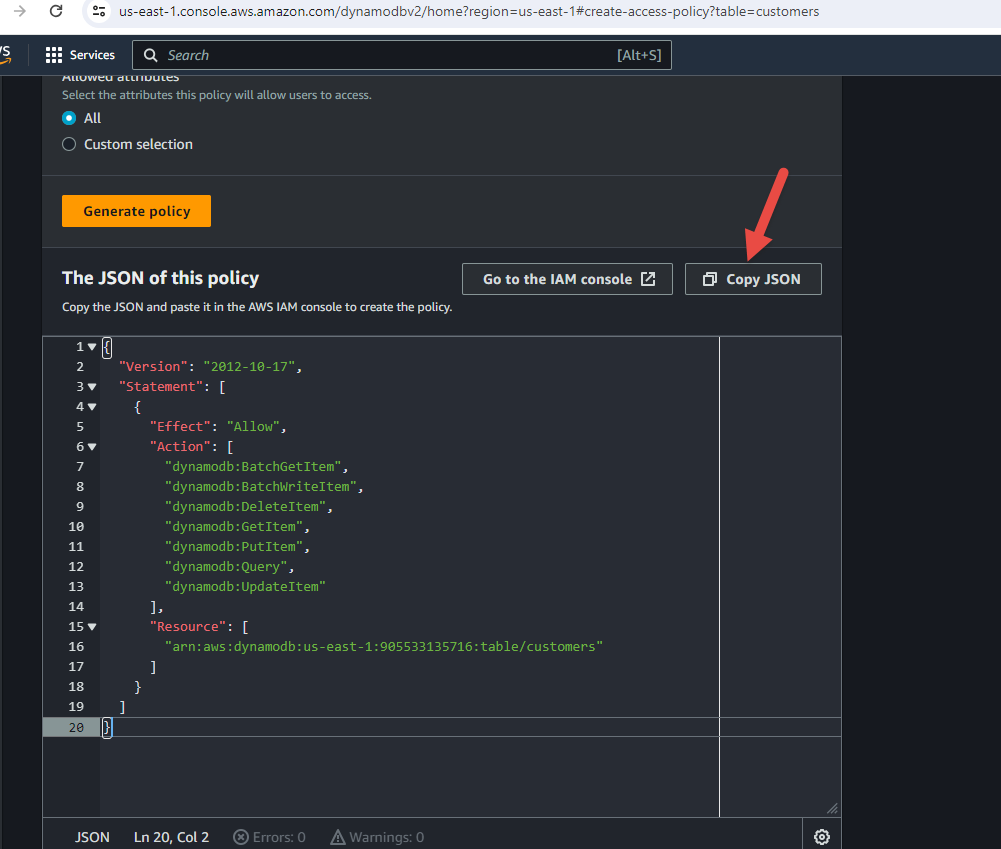


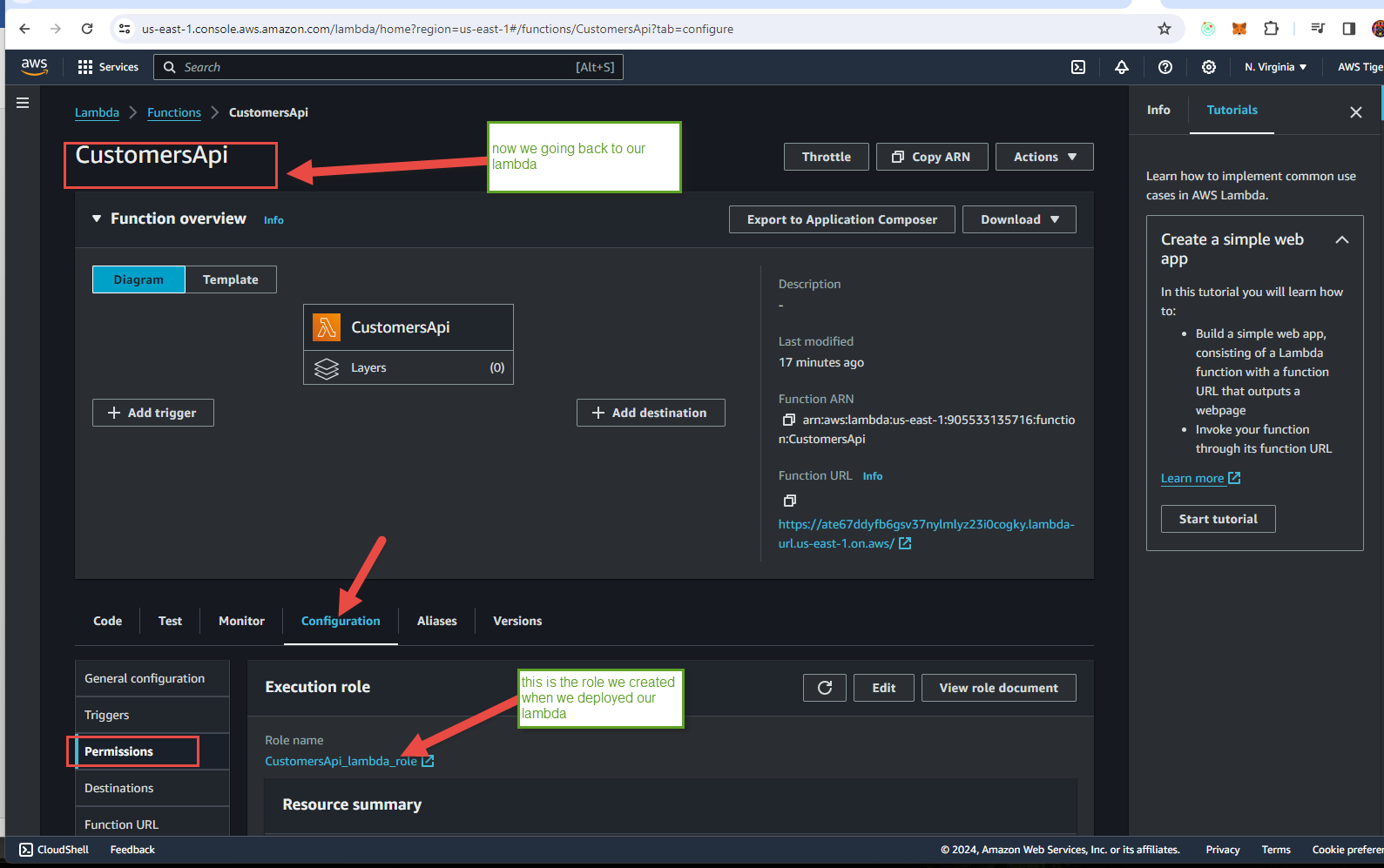
Now after we deploy our lambda in AWS we need to provide the access for our lambda to be able to communicate with our dynamo db customers. This is what we need to do:

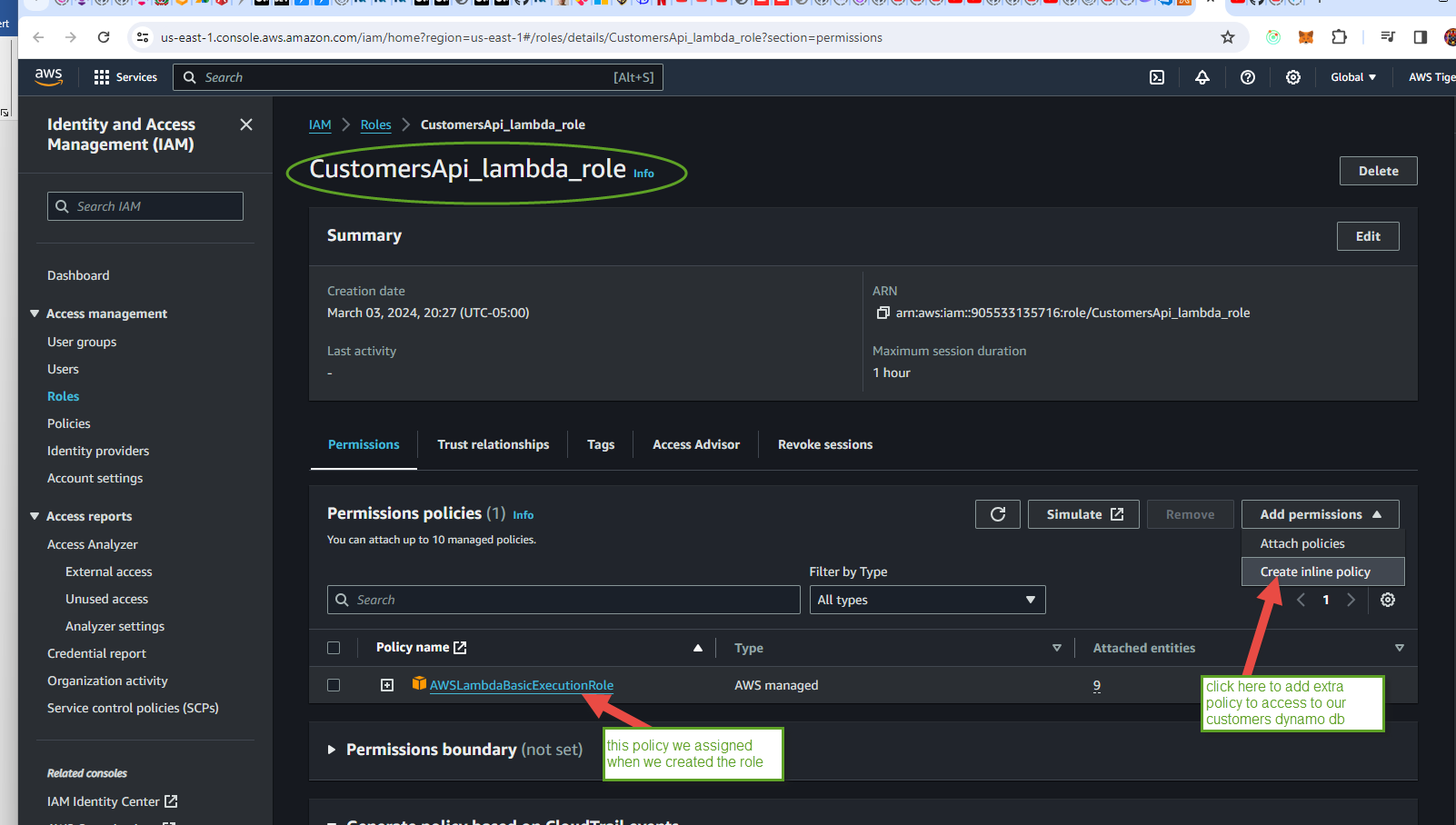


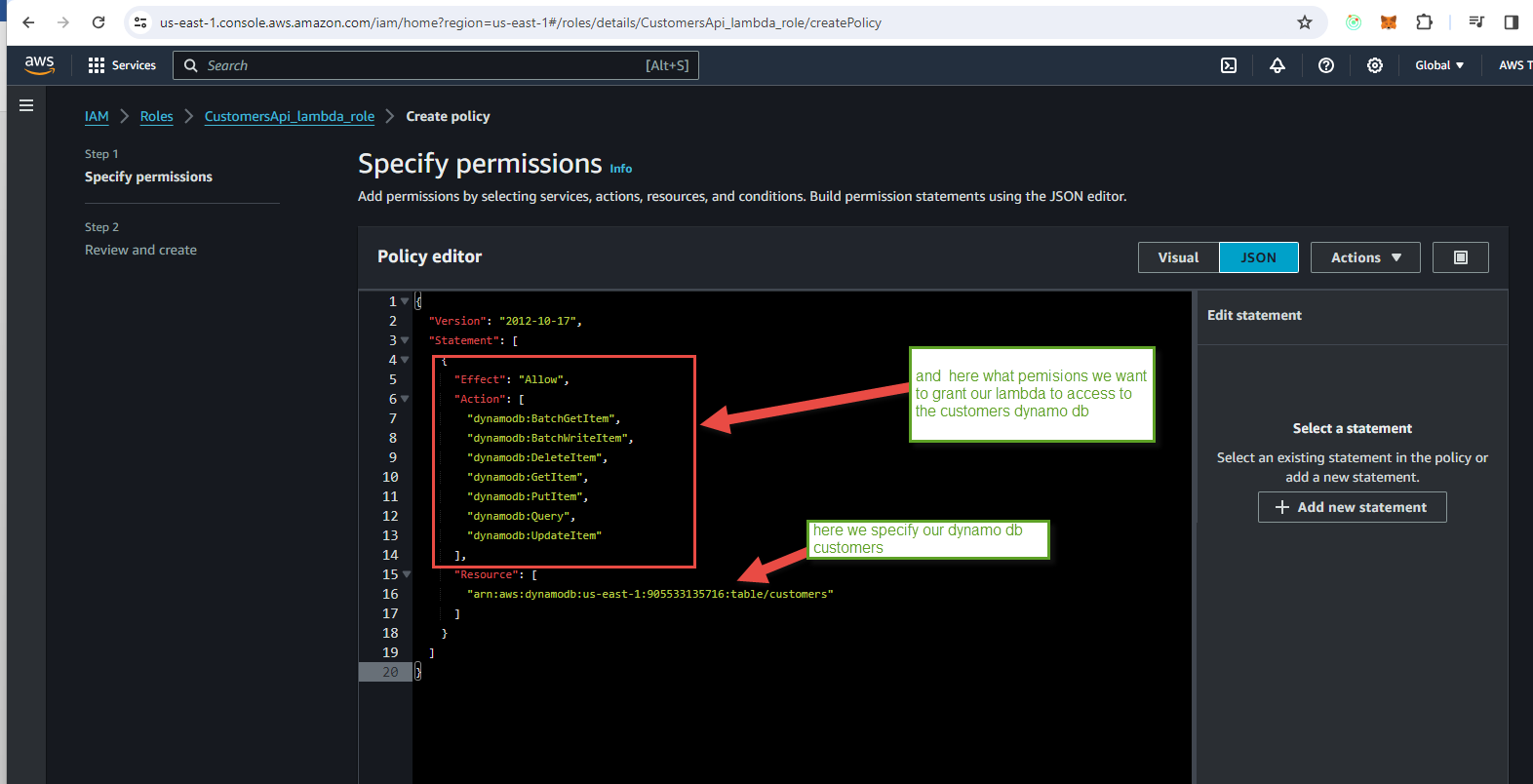


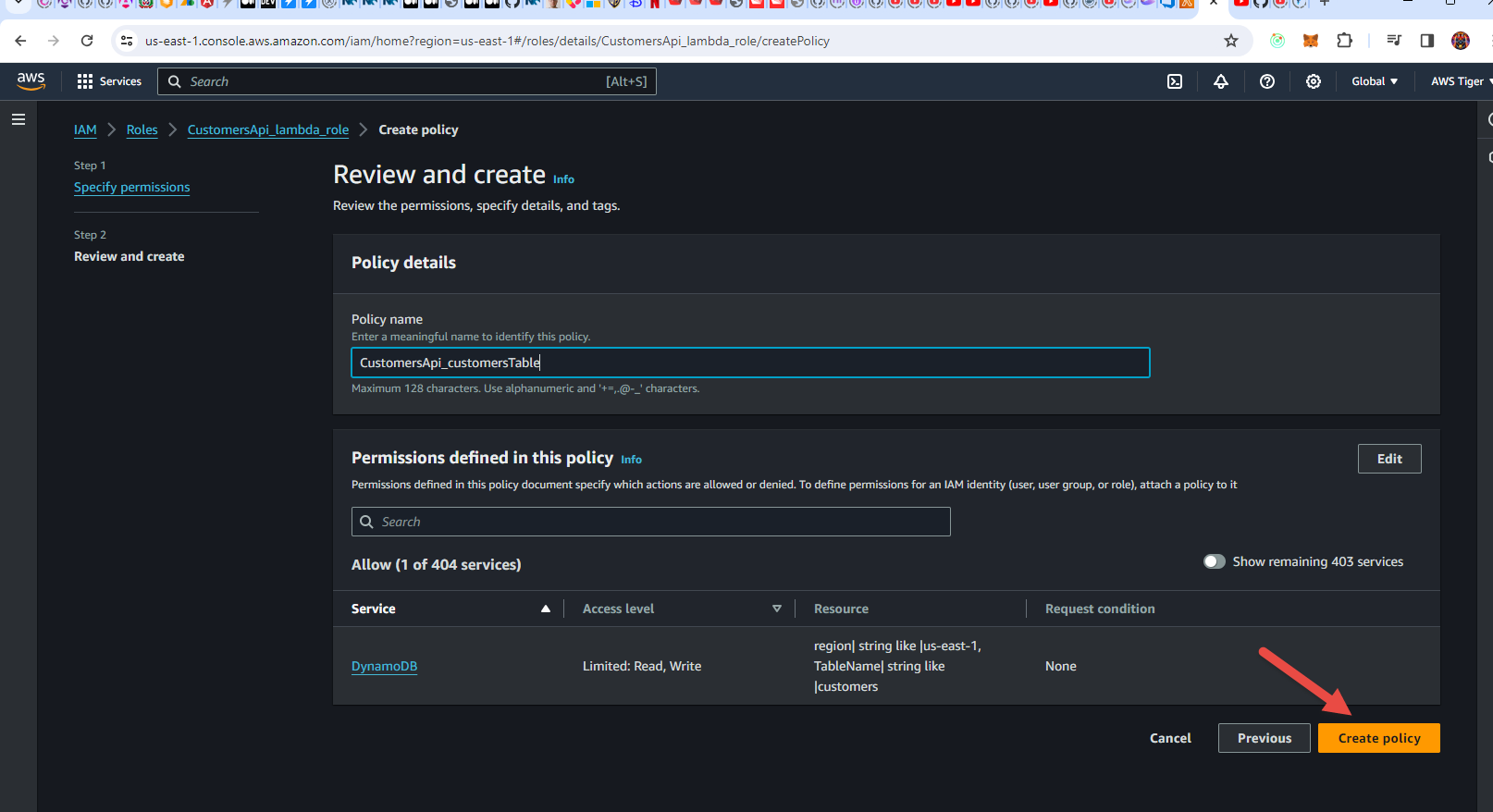


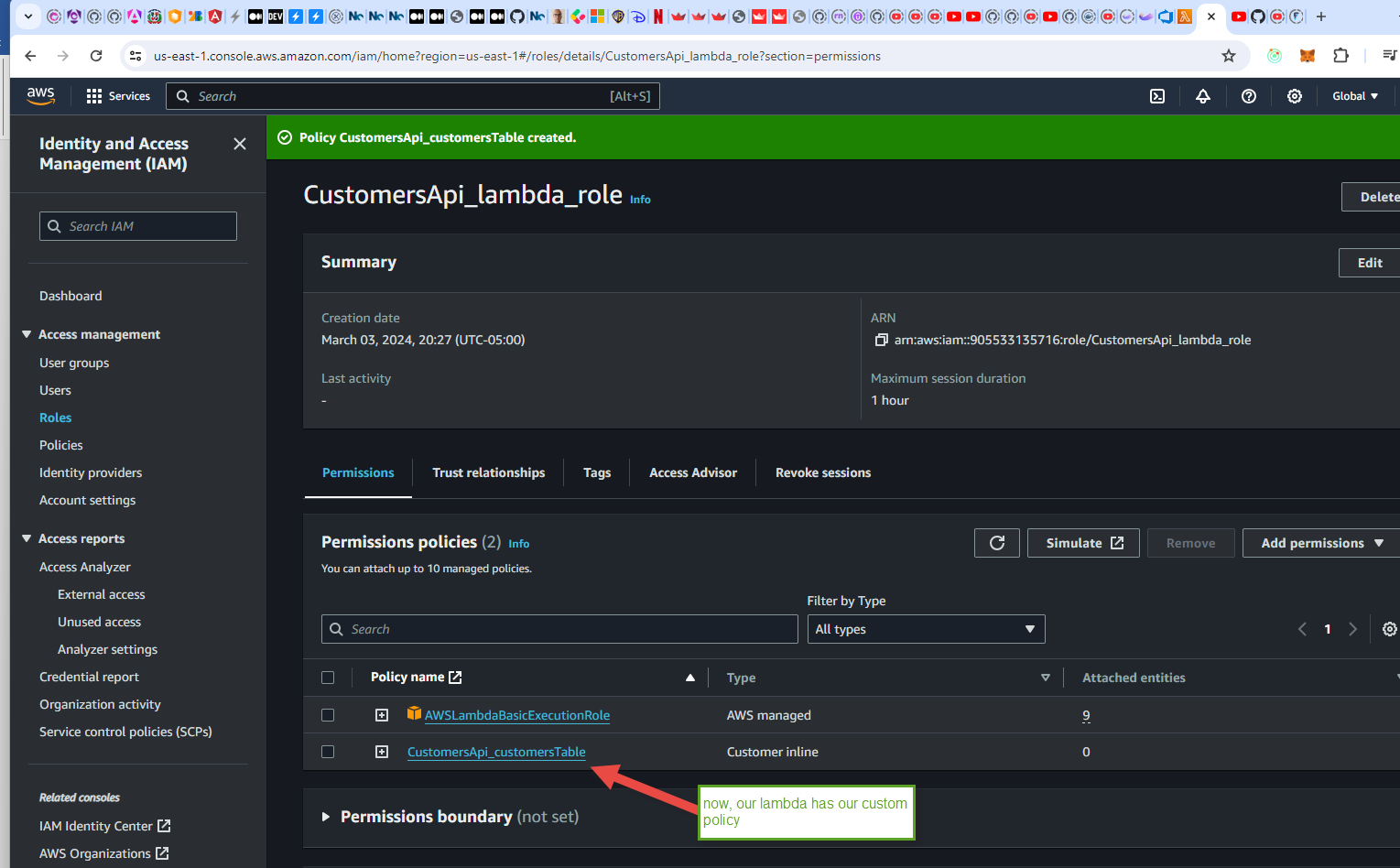




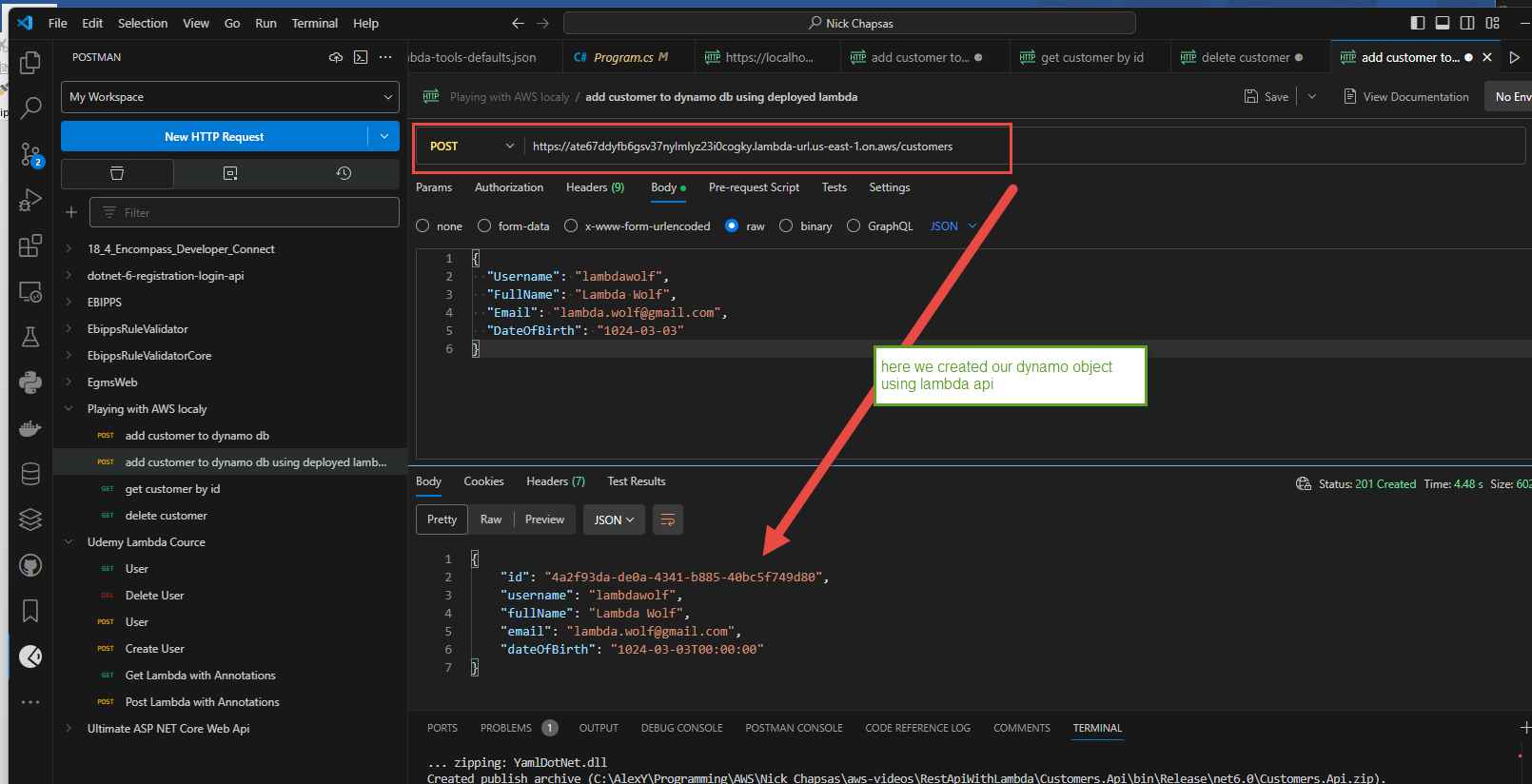








Now we are ready to test our lambda in Postman again, but this time we are using our lambda service url from AWS:



And retrieve:

