Create your AWS account

You can just search for "aws create account" in your browser.
Here's a quicker link: https://aws.amazon.com/resources/create-account/

After creating your account, redirect to Amazon Sagemaker(you can type in the search bar to locate faster), and click on "Studio" on the left side bar.

Create the endpoint

Follow the instruction in the link: https://aws.amazon.com/blogs/machine-learning/llama-2-foundation-models-from-meta-are-now-available-in-amazon-sagemaker-jumpstart/, and deploy the chosen model.

Notice the endpoint may require you to run on particular instances, for Llama-2-7b-chat, you will likely to receive the following message:

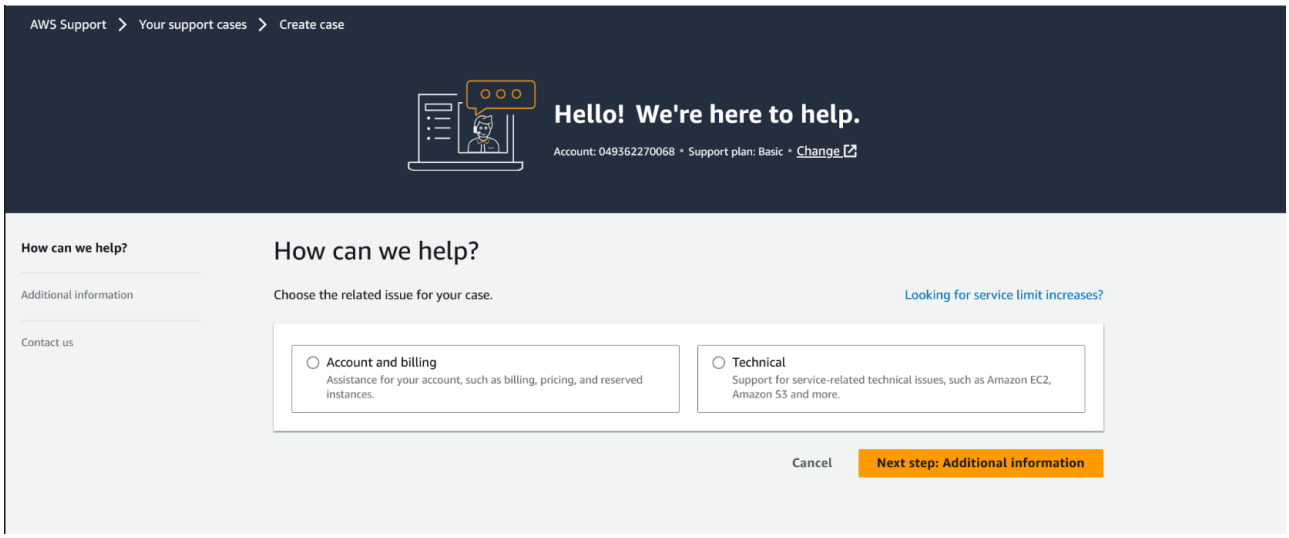
" We encountered an error while preparing to deploy your endpoint. You can get more details below.

An error occurred (ResourceLimitExceeded) when calling the CreateEndpoint operation: The account-level service limit 'ml.g5.2xlarge for endpoint usage' is 0 Instances, with current utilization of 0 Instances and a request delta of 1 Instances. Please use AWS Service Quotas to request an increase for this quota. If AWS Service Quotas is not available, contact AWS support to request an increase for this quota."

Request instance

In order to use a compute enhanced notebook instance, you must submit a request for a service limit increase to the AWS Support Center.

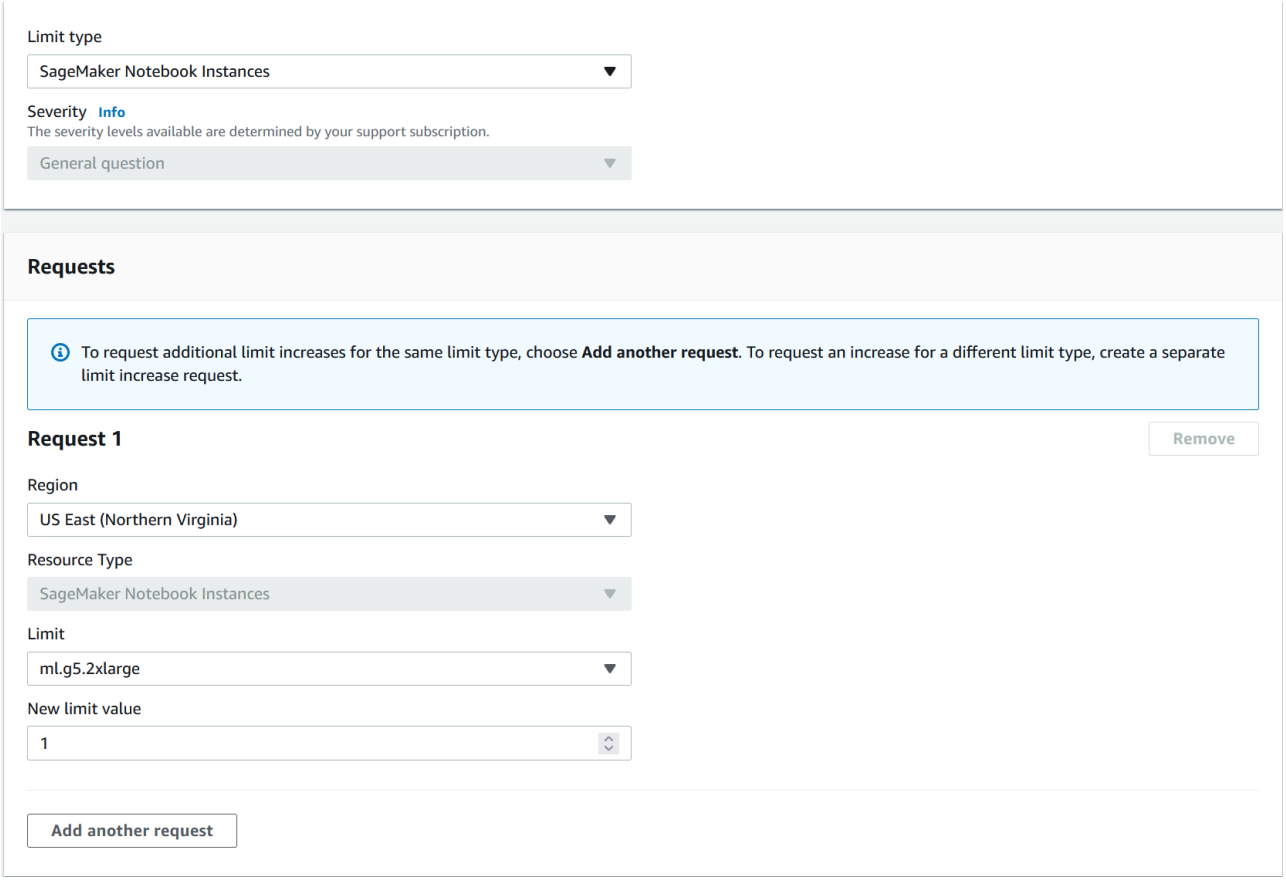
1. Open the AWS Support Center console.
2. On the AWS Support Center page, choose Create Case and then choose Looking for service limit increases?



3. In the Limit type panel, search for SageMaker Notebook Instances.
4. In the Request panel, choose the Region that you are working in(US East (Northern Virginia)). For Resource Type, choose SageMaker Notebook.

5. For Limit choose ml.g5.2xlarge instances.

6. For New Limit Value, verify that the value is 1.



7. In Case description, provide a brief explanation of why you need the Service limit increase. For example, I need to use this to deploy Llama-2-7b-chat.

8. In Contact options, provide some details about how you would like to be contacted by the AWS service support team on the status of your Service limit increase request.

9. Choose submit.

PS: I followed the instruction in th link: https://docs.aws.amazon.com/deepcomposer/latest/devguide/deepcomposer-service-limit.html amd submitted case. I was later contacted by Amazon saying I "submitted for the wrong resource type. The error you provided mentions the CreateEndpoint operation, but the request was submitted for Notebook Instance resources (CreateNotebookInstance operation)." Of course through communication, they assigned me the quota. I did not attempt with submitting "CreateEndpoint" operation as the resource type, but anyone is encouraged to try this.

As long as you deploy the model in Sagemaker Studio, which will take a while, you should be able to open a notebook like the one below, beginning from section "Run inference on the Llama 2 endpoint you have created." to "Query endpoint that you have created", including all major contents, along with some of our edits.

Something to note

The script expects that you have created a SageMaker endpoint named "jumpstart-dft-meta-textgeneration-llama-2-7b-f"(or any name you have on your end) and that you have appropriate access rights to access the

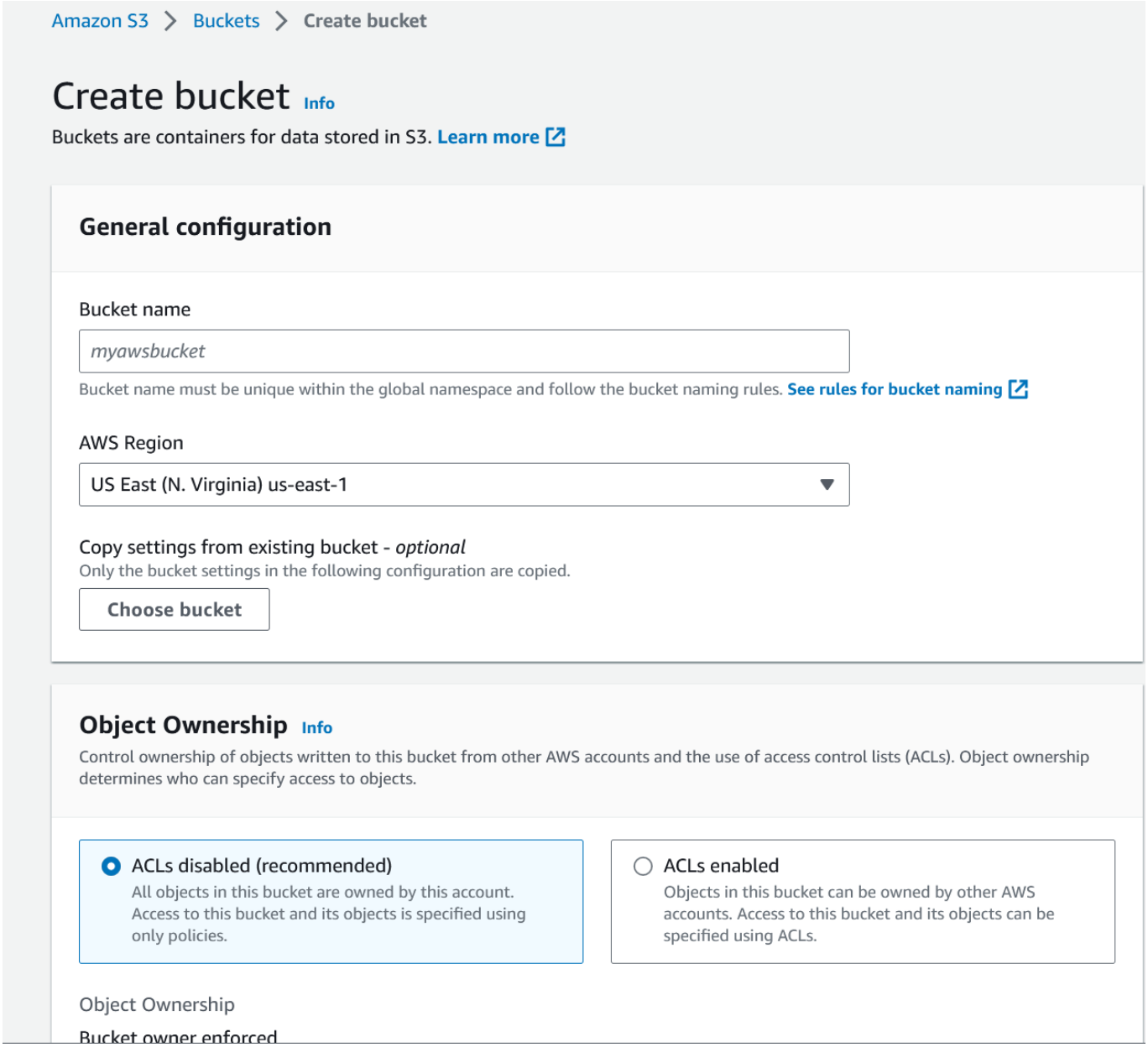
S3 bucket and perform actions on the SageMaker endpoint. Keep in mind that this script uses a specific Llama 2 model with predefined parameters. If you want to use a different model or set different parameters, you can modify the payload dictionary within the loop.

Sagemaker auto-generated endpoint notebook

You should be able to open a notebook after deploying the endpoint. Here's ours for reference: meta-textgeneration-llama-2-7b-f

Run inference on the Llama 2 endpoint you have created.

If you wish to read files or data, you will need to create a bucket in S3:



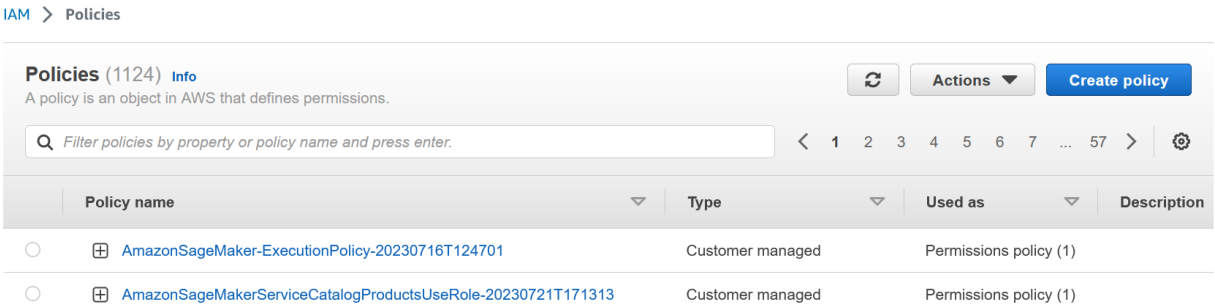
After creating a bucket, you can upload desired files and datasets.
Here's the link for more info on buckets:
https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html?icmpid=docs_amazons3_console

API Call Using Amazon API Gateway and AWS Lambda

Resources for API call (tested on postman)

To allow an API call, one needs to build an API gateway and a lambda function to invoke the endpoint then return to the user.
We referenced from this blog: https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/

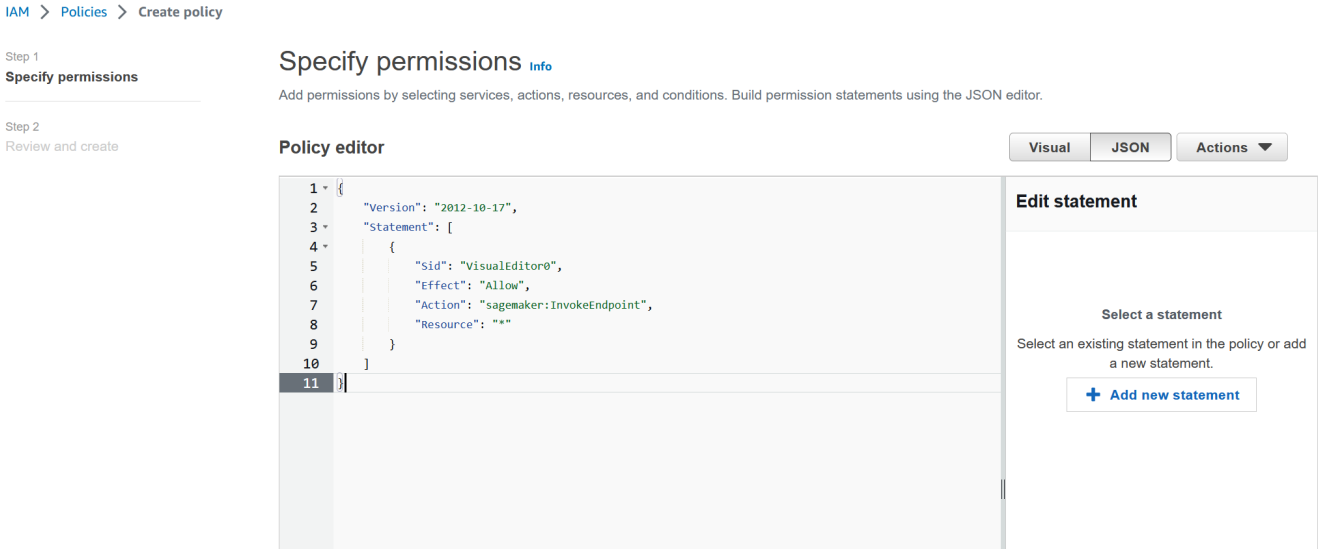
1. Create a role that allows lambda function to invoke the sage maker endpoint Direct to IAM policies and create a new poilcy. Select **Create policy**.



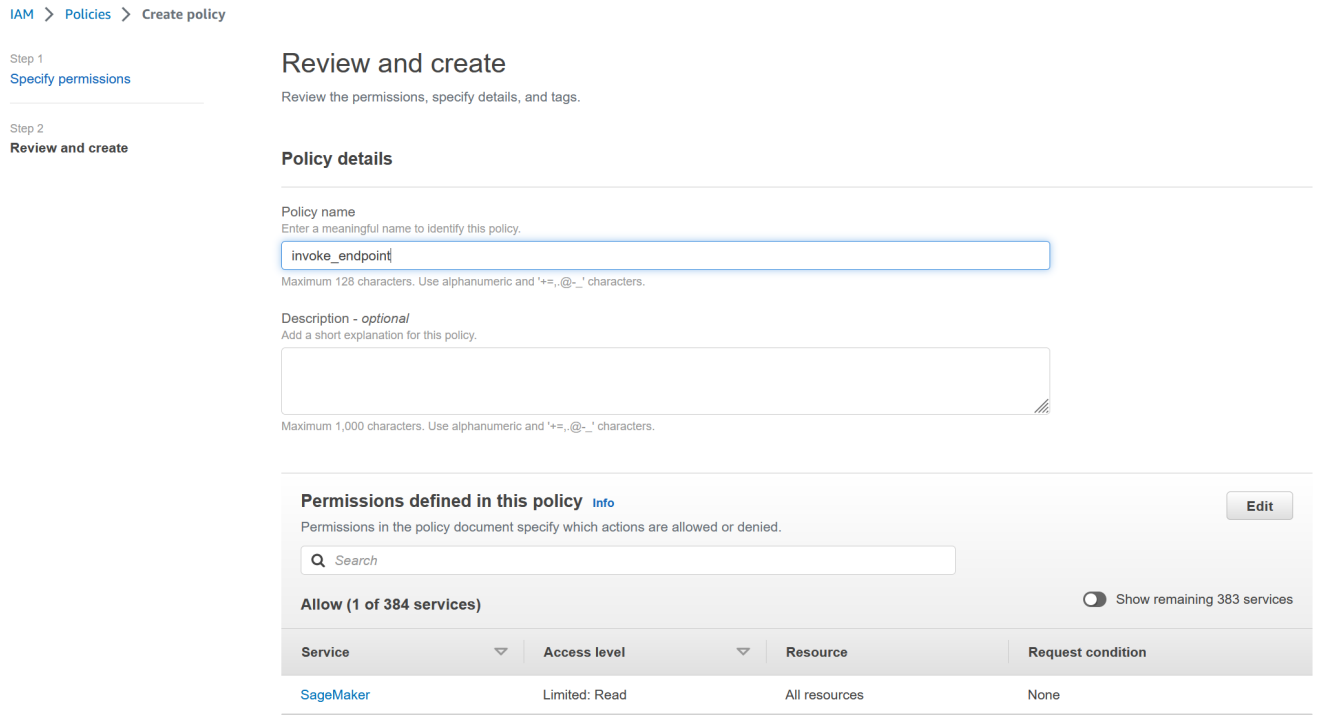
Swtich to **JSON** and copy the following code as your policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "sagemaker:InvokeEndpoint",
            "Resource": "*"
        }
    ]
}
```

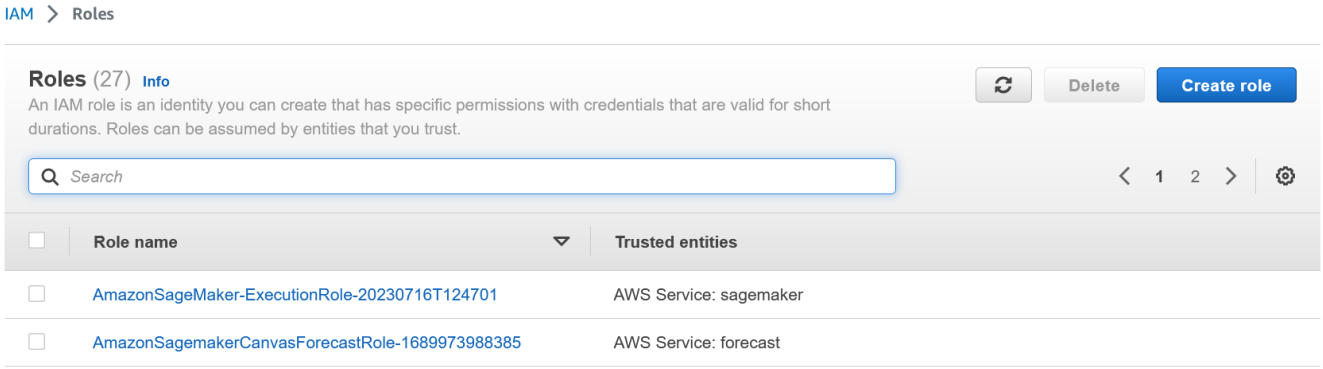
You shall see the page like this:



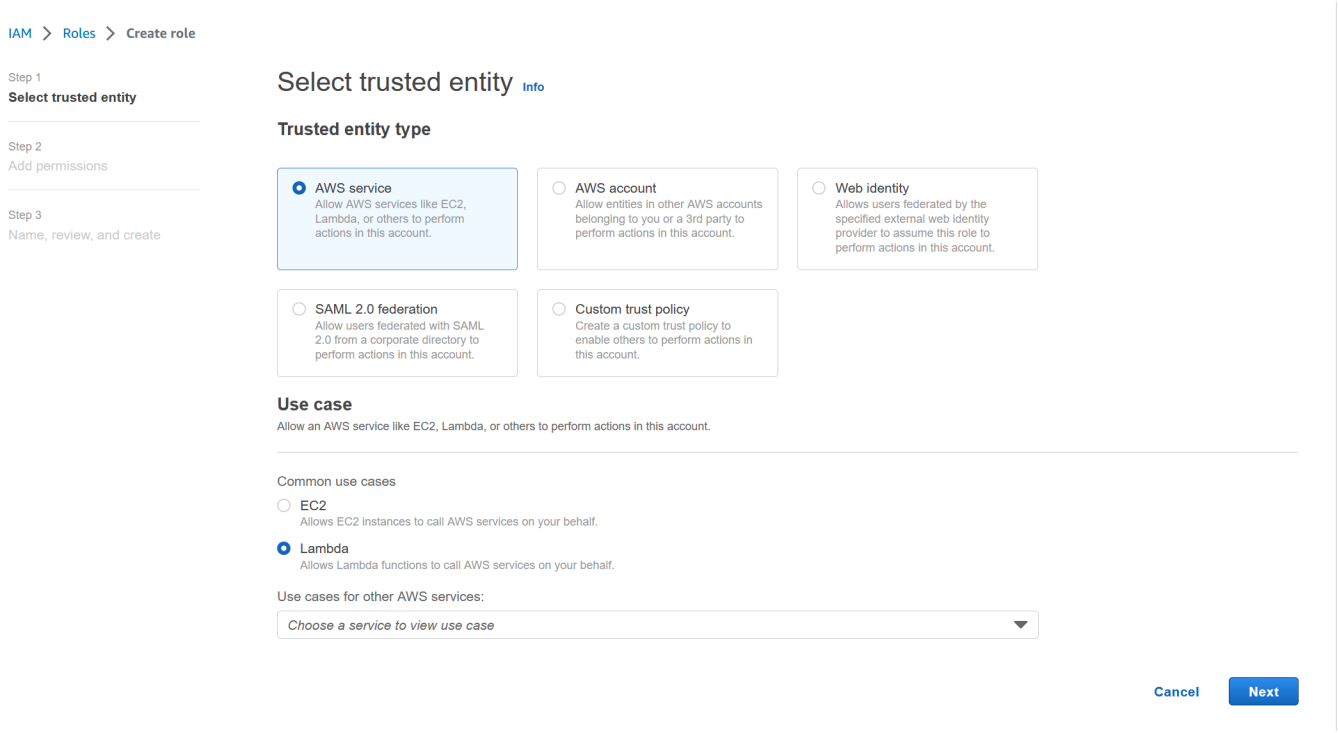
Click 'Next' and name your policy then click **Create policy** to finish.



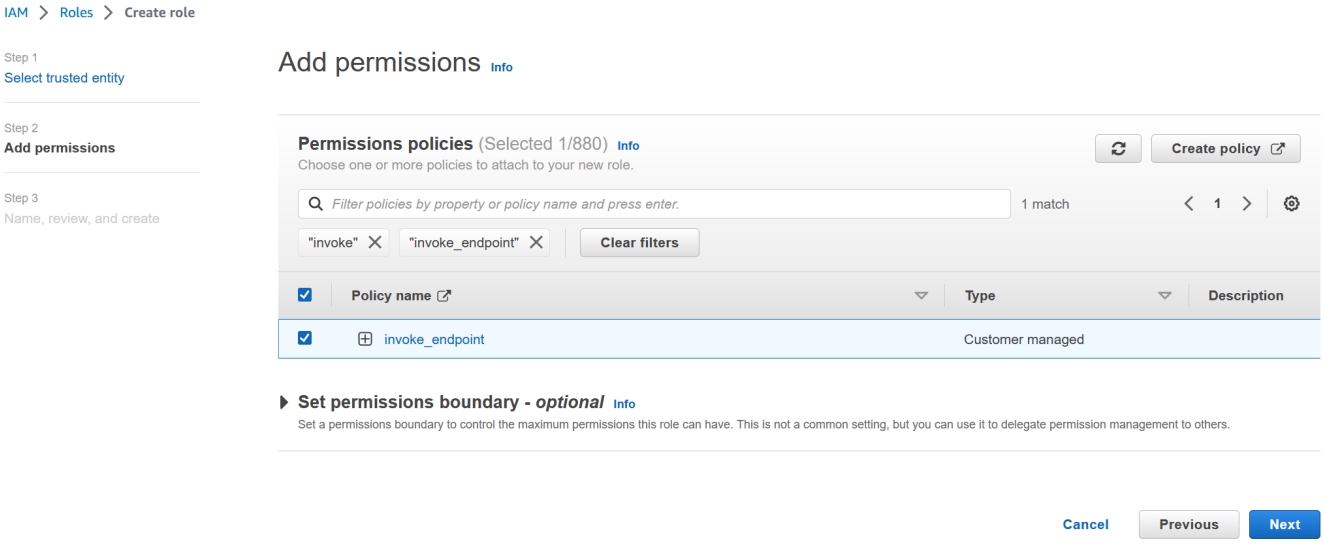
Now we can create a role using the new policy. Direct to IAM Roles and click **Create role**.



We selected Lambda as our service. Click **Next** when you are done.

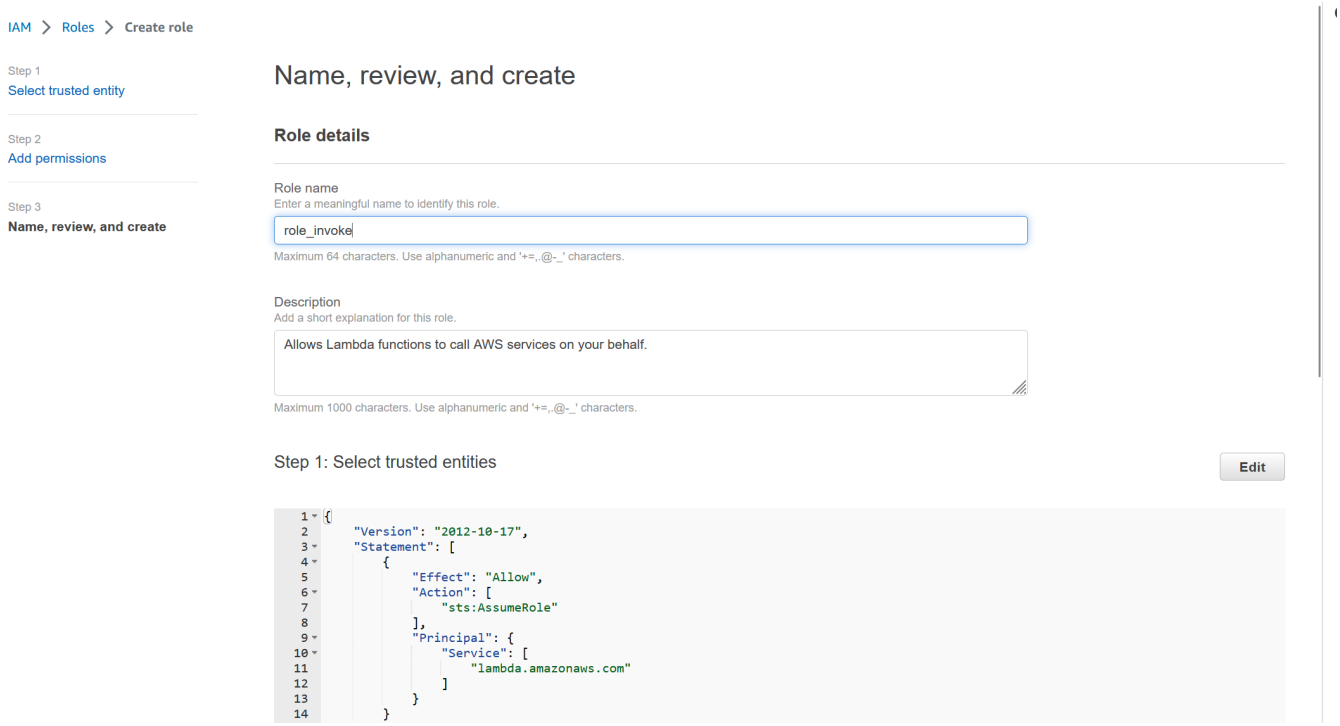


You shall see a list of all policies. Select the one you just created. You can search for a quicker result.



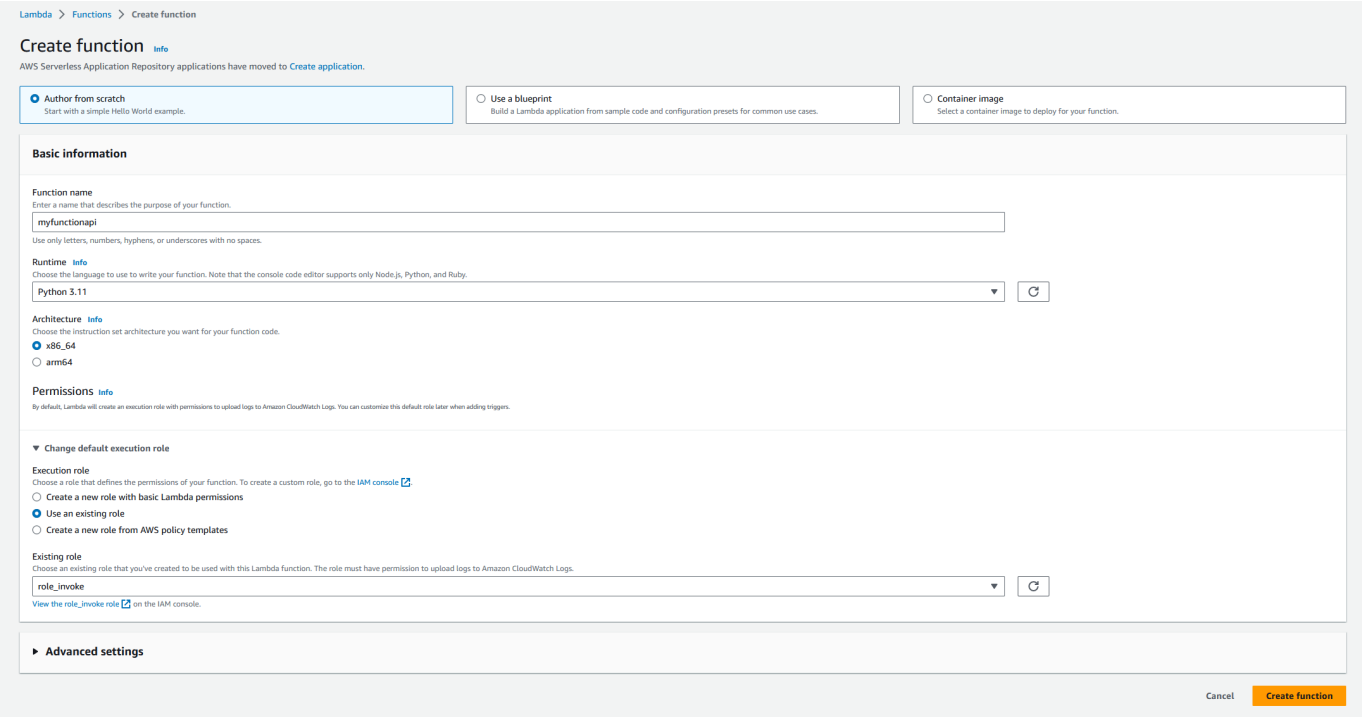
Click **Next** when you are finished.

Name your role and you shall see the policy you added.

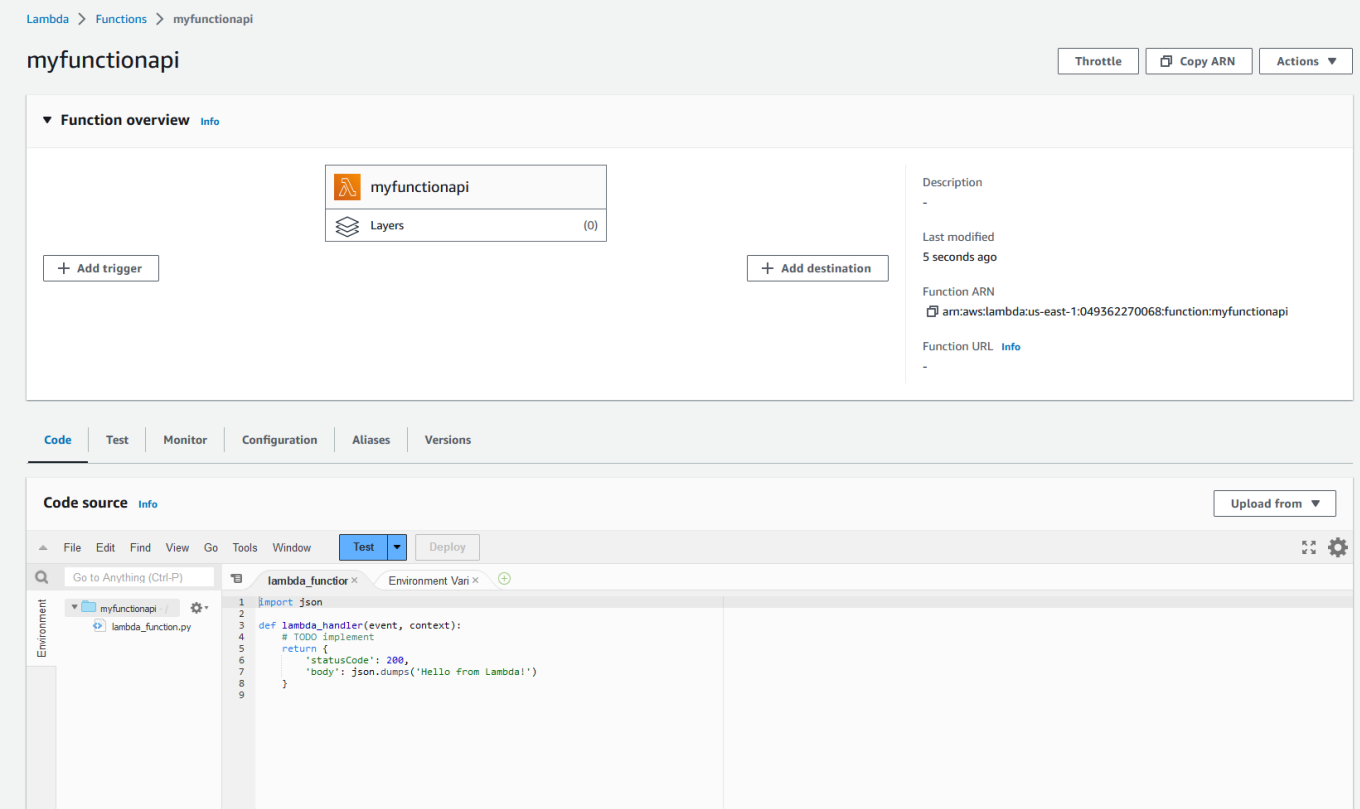


2. Create a Lambda function

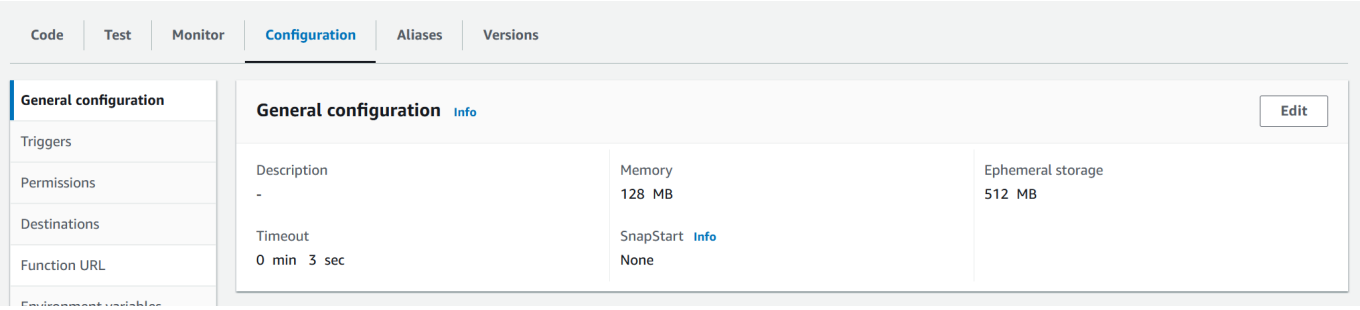
Create a new function and select function language. We used Python 3.11 in our case.
In **Change default execution role**, select **Use an existing role** and add your role.



After creating the function, you shall see a page like this:

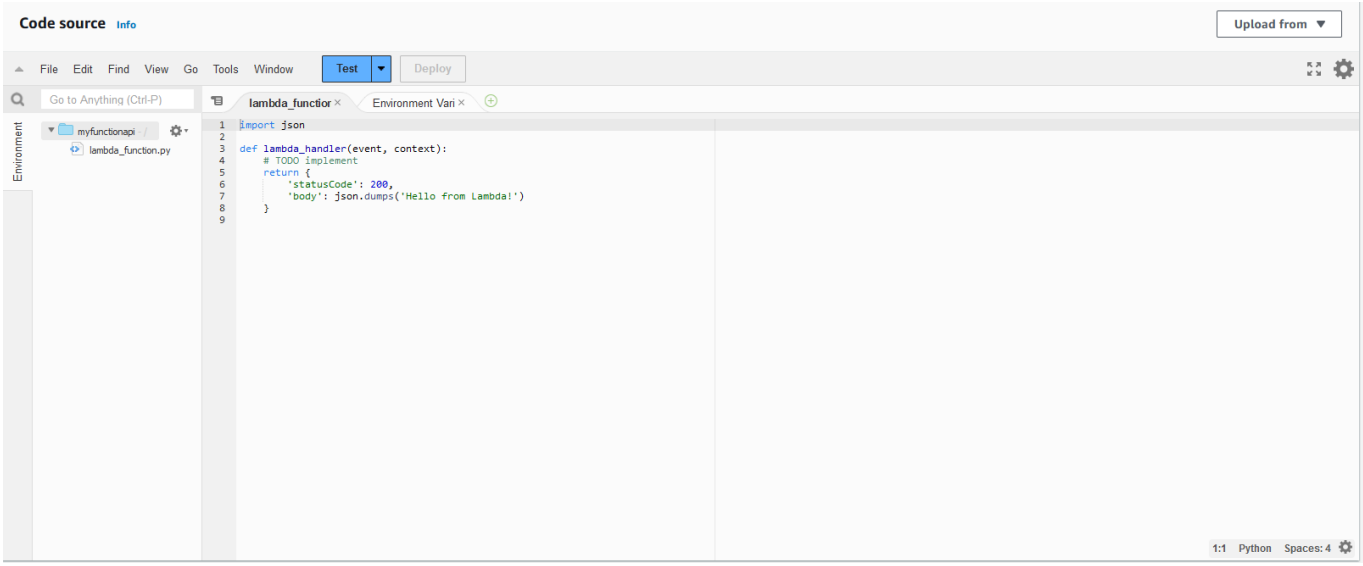


You can edit your function in **Code** section and configure more details using **Configuration**. For example, the default timeout is 3 seconds, which might not be enough in your case and will likely return an error. You can increase the length in **Configuration** and edit the preferrable timeout.



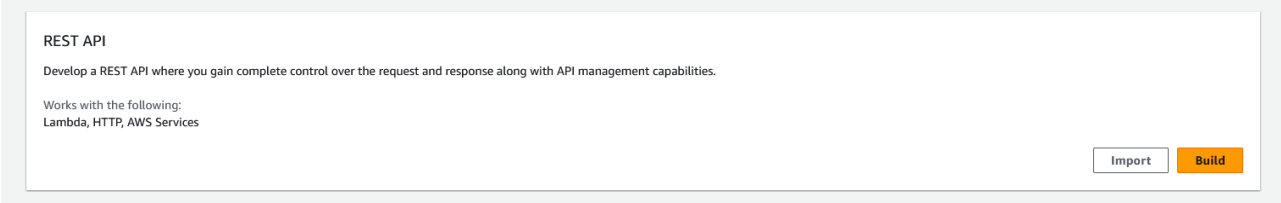
Here are more info about Lambda: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html?icmpid=docs_lambda_help

When you make any changes, make sure to click **Deploy** to save the changes. You can also create test case(s) to examine your code.

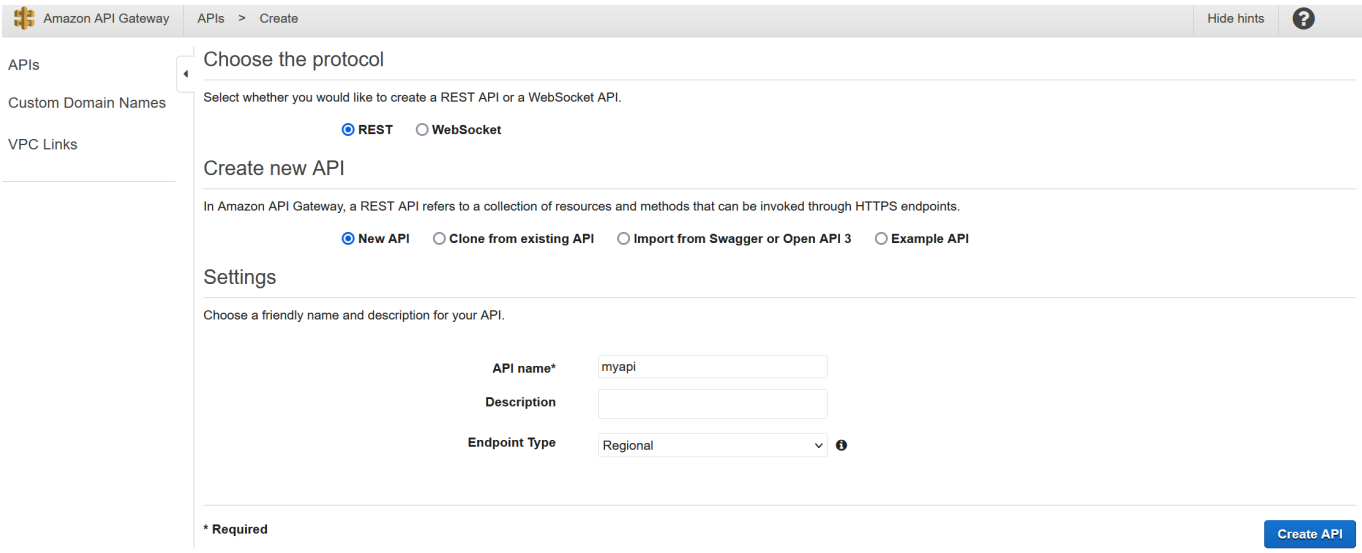


Here's a quick access to our function: Lambda_Function

3. Create a REST API Direct to API Gateway and select **Build** of REST API.

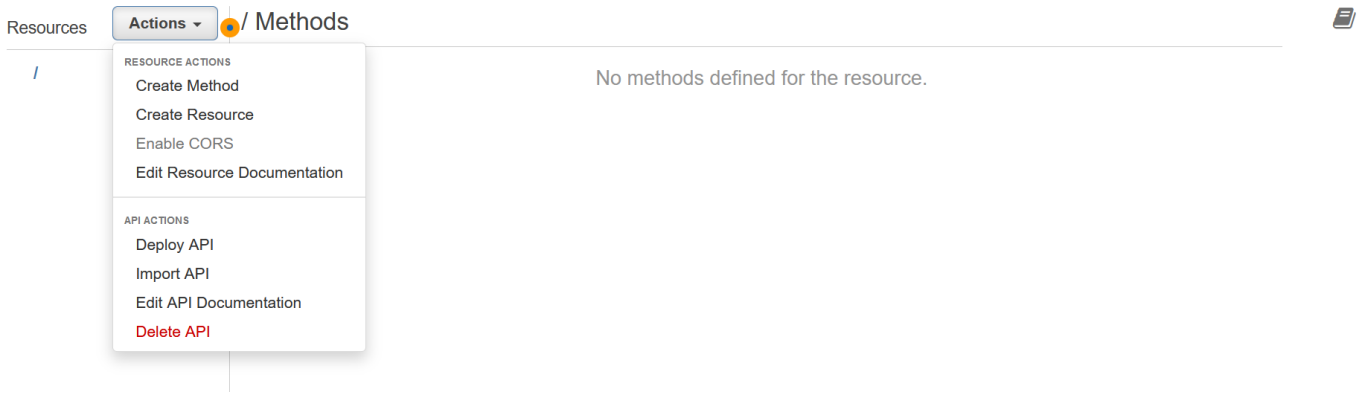


Select accordingly and name your API. Click **Create API** when you are done.

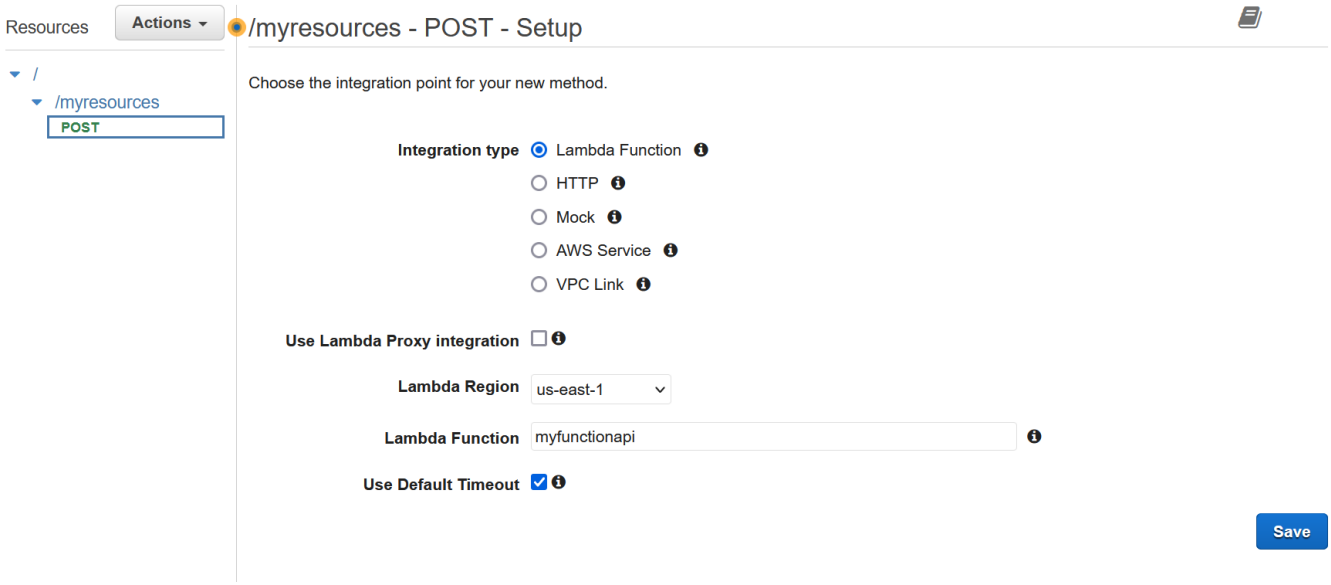


On the **Actions** menu, choose **Create resource**. Enter a name for the resource. After the resource is created, on the **Actions** menu, choose **Create Method** to create a method (It could be any method depends on your

need. We created POST).

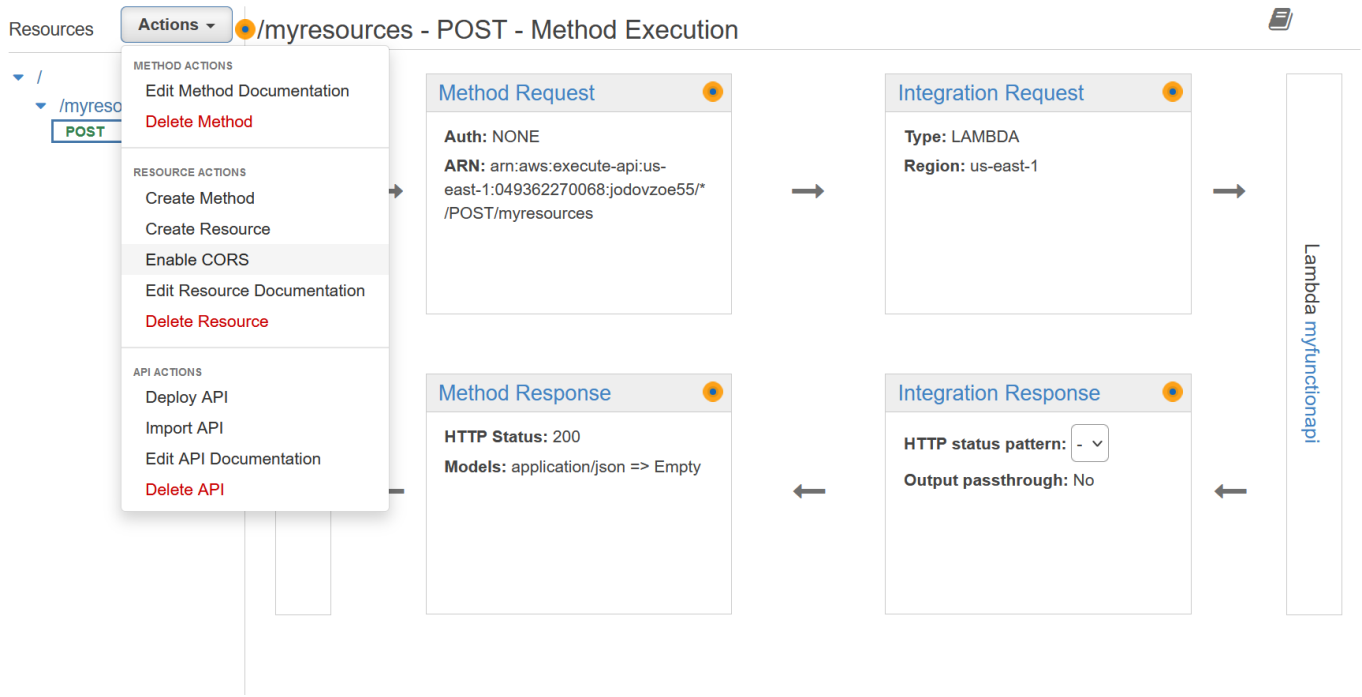


When you created the resource and method, you shall see the following content:

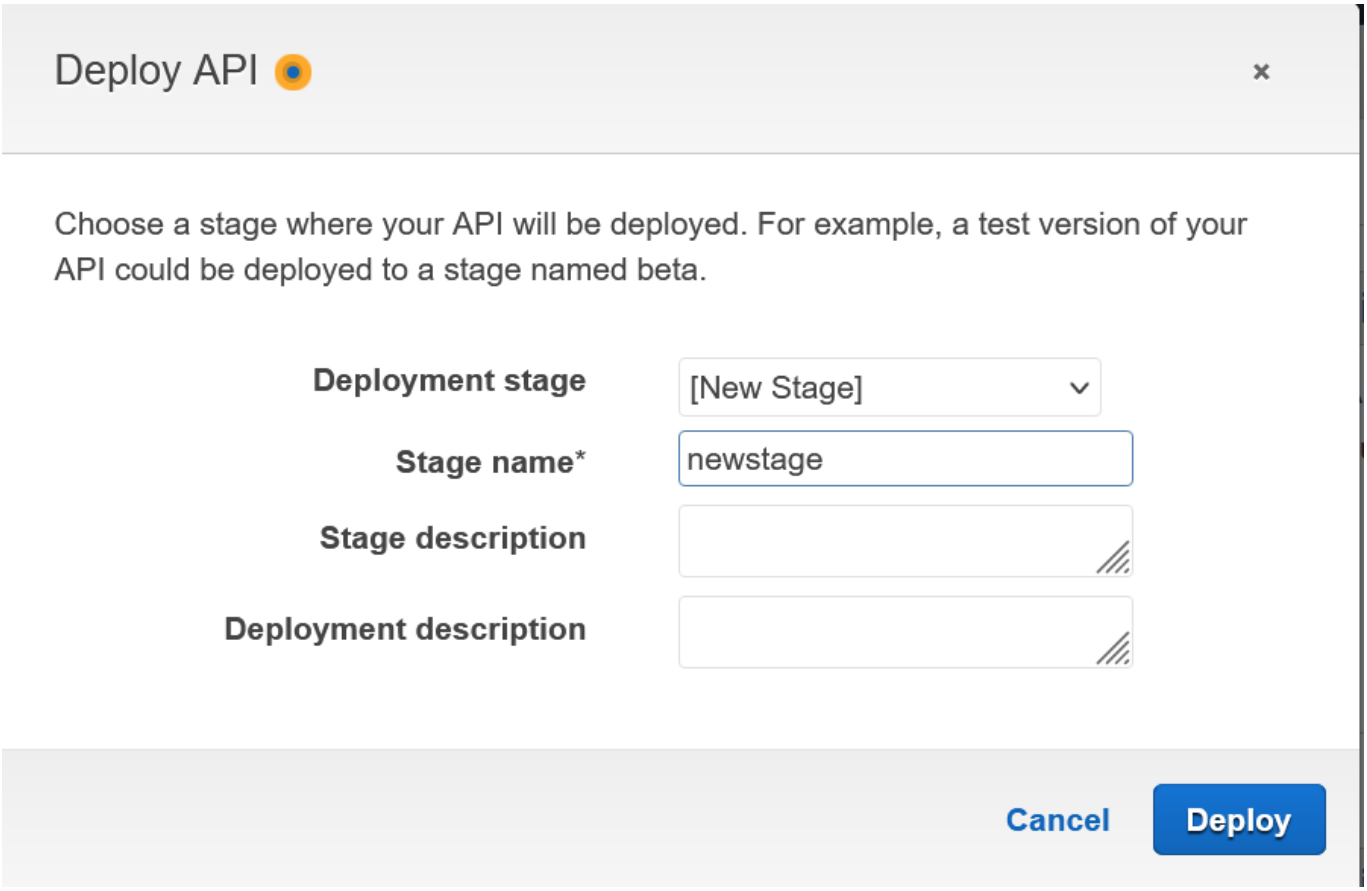


Enter your Lambda function and click **Save**.

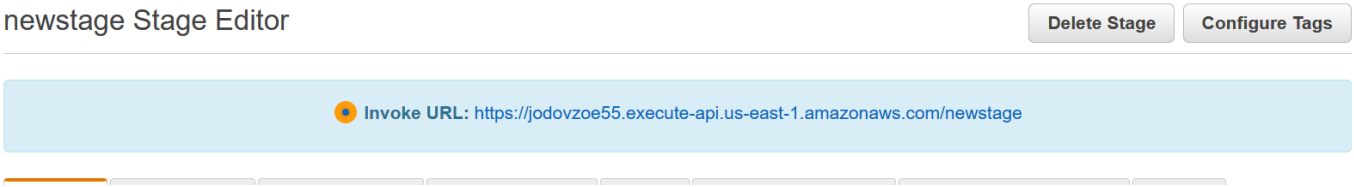
Now we can deploy the API through **Actions - Deploy API**:



One last step, name your stage and hit **Deploy** when you are ready.

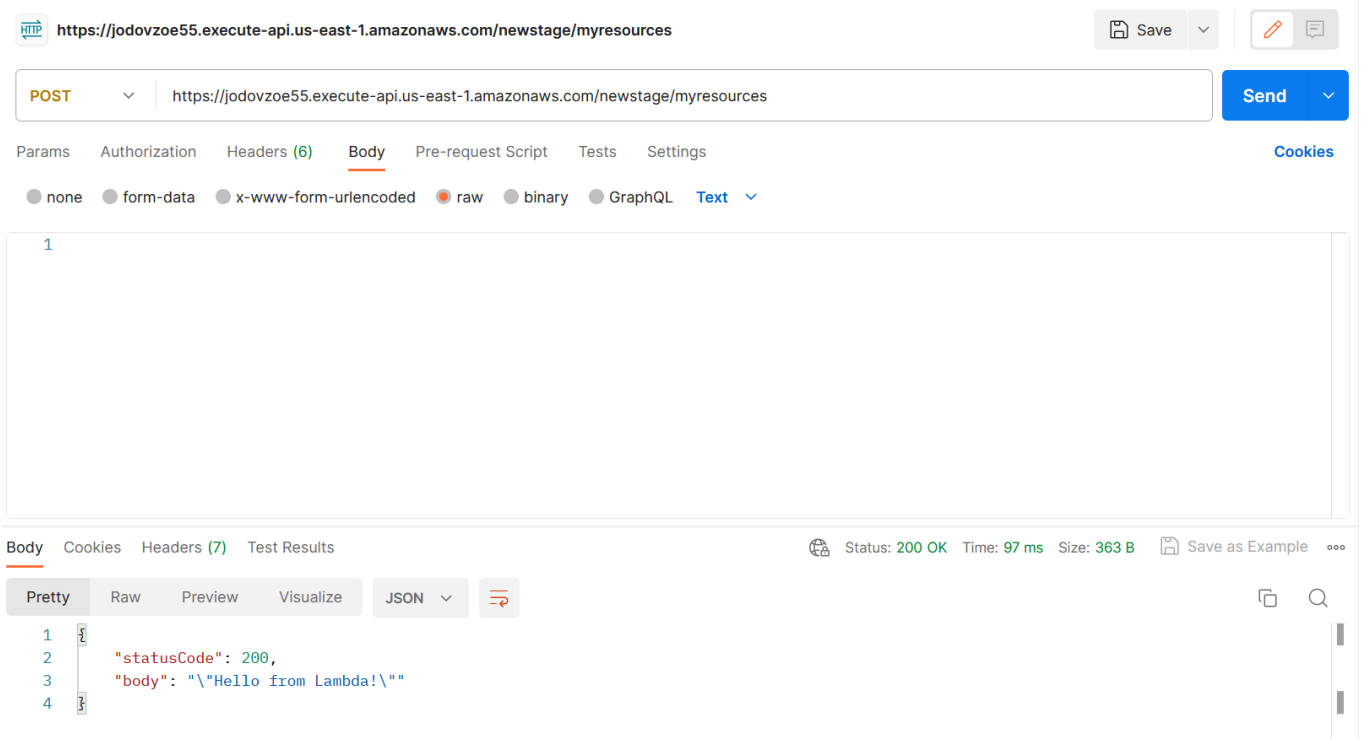


You will receive your invoke url:



4. Test in Postman

You input url should follow the format: `https://{restapi_id}.execute-api.{region}.amazonaws.com/{stage_name}/{resource_name}` In our case, we select **POST** as the method and enter our complete url, with the result returned below:



If your lambda function aims to take inputs and return outputs, you can test by input data in **Body - raw**.

More references

https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints-test-endpoints.html
https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_runtime_InvokeEndpoint.html
https://aws.amazon.com/blogs/machine-learning/llama-2-foundation-models-from-meta-are-now-available-in-amazon-sagemaker-jumpstart/

https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started.html

Clean up

When using AWS, a platform both powerful and expensive, you want to clean up all your running resources to avoid accidental charges. Following the guidlines through this link: https://docs.aws.amazon.com/sagemaker/latest/dg/ex1-cleanup.html, in additional with Step 6 in another link: https://aws.amazon.com/tutorials/machine-learning-tutorial-deploy-model-to-real-time-inference-endpoint/. Then you should be able to shut down all running resources. If you want to revisit some resources later, you are welcome to keep them, but be aware this could incur accumulating charges.

Usually the free tier does have free usuage for Lambda Function and API, but you may want to delete them to prevent charges and clean up some space. Here's a guideline: https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started.html

To monitor your bills, you can setup alerts as preventive measures:

-- Budgets: http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/budgets-managing-costs.html
-- CloudWatch billing alerts and alarms: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html