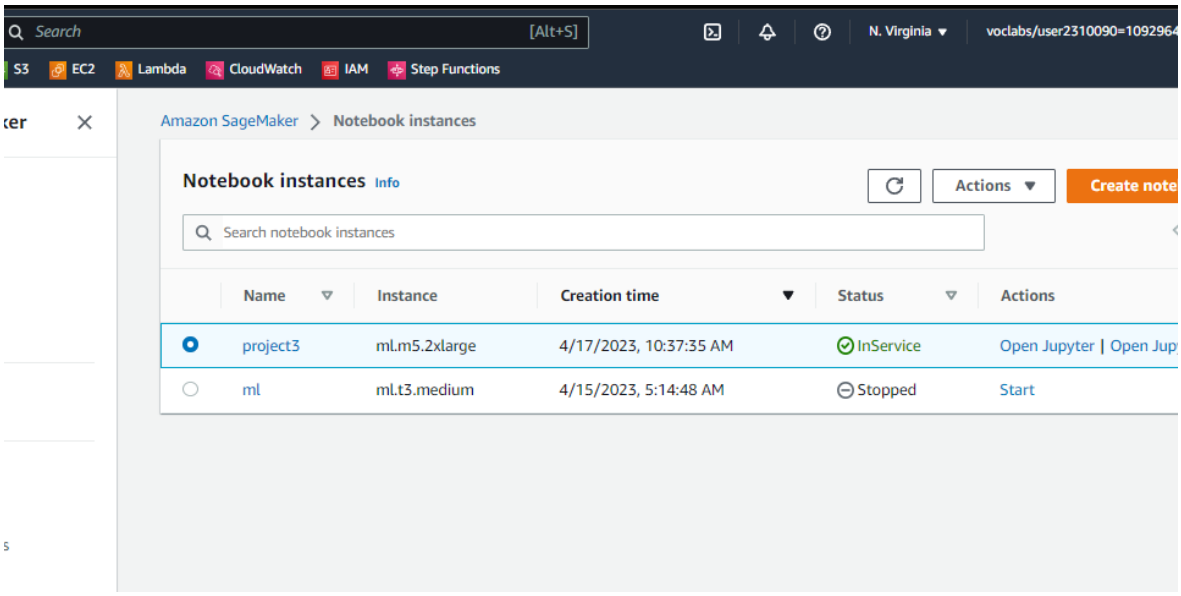
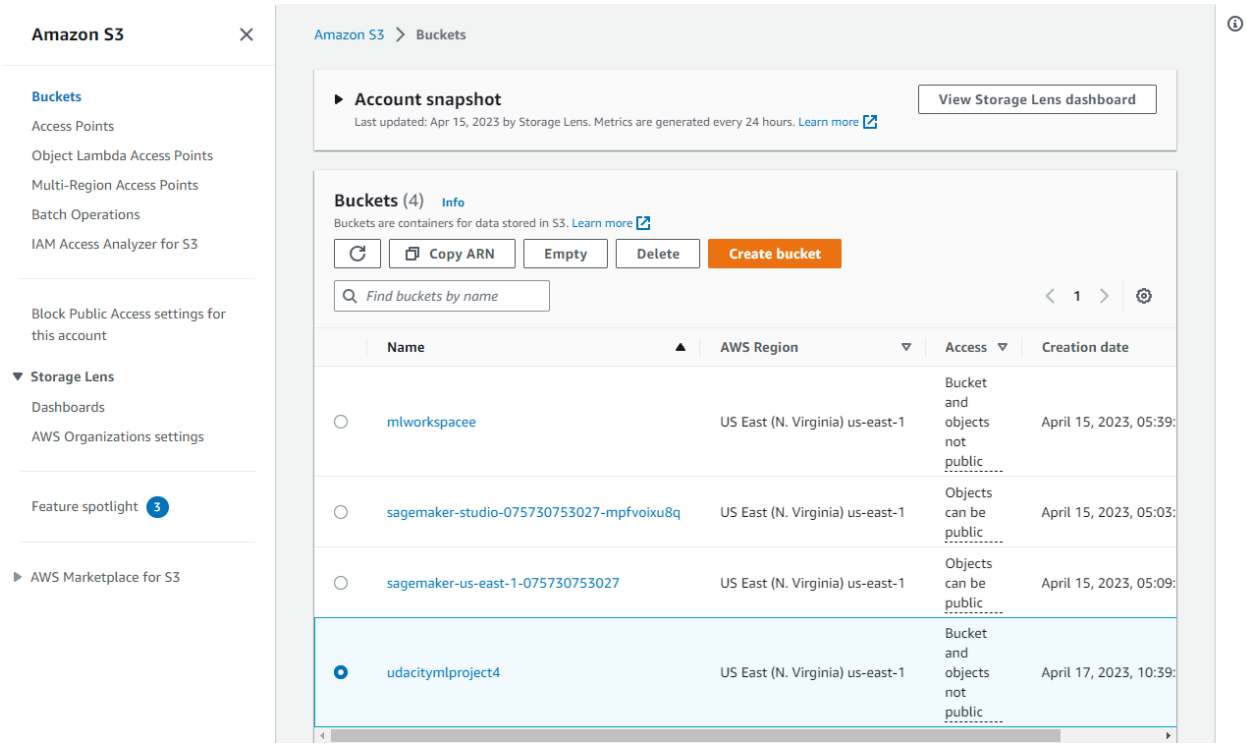
Operationalizing an AWS ML Project

Step 1: Training and deployment on Sagemaker

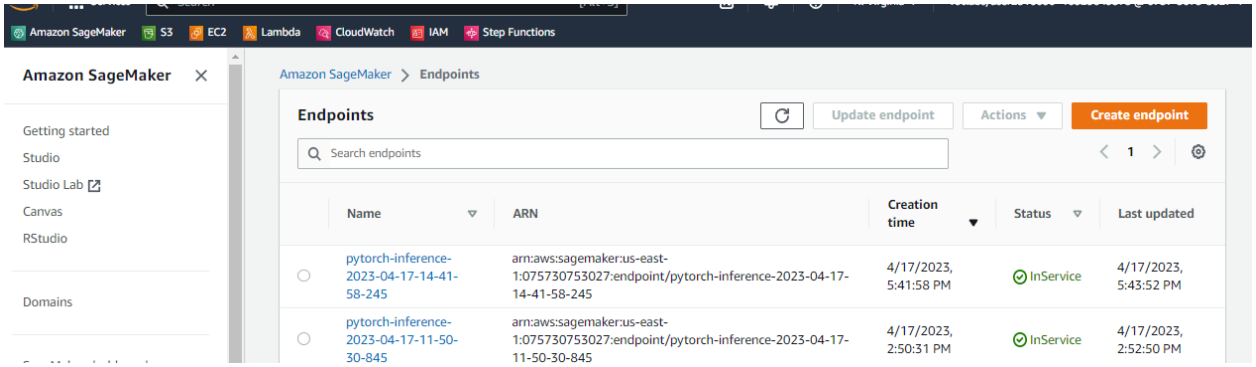
- Selecting the notebook: I have started by using **ml.t3.medium** for the notebook instance. But it took a long time to process image files and got stuck at some place. So I have changed to **ml.m5.2xlarge**



- S3 Bucket Creation

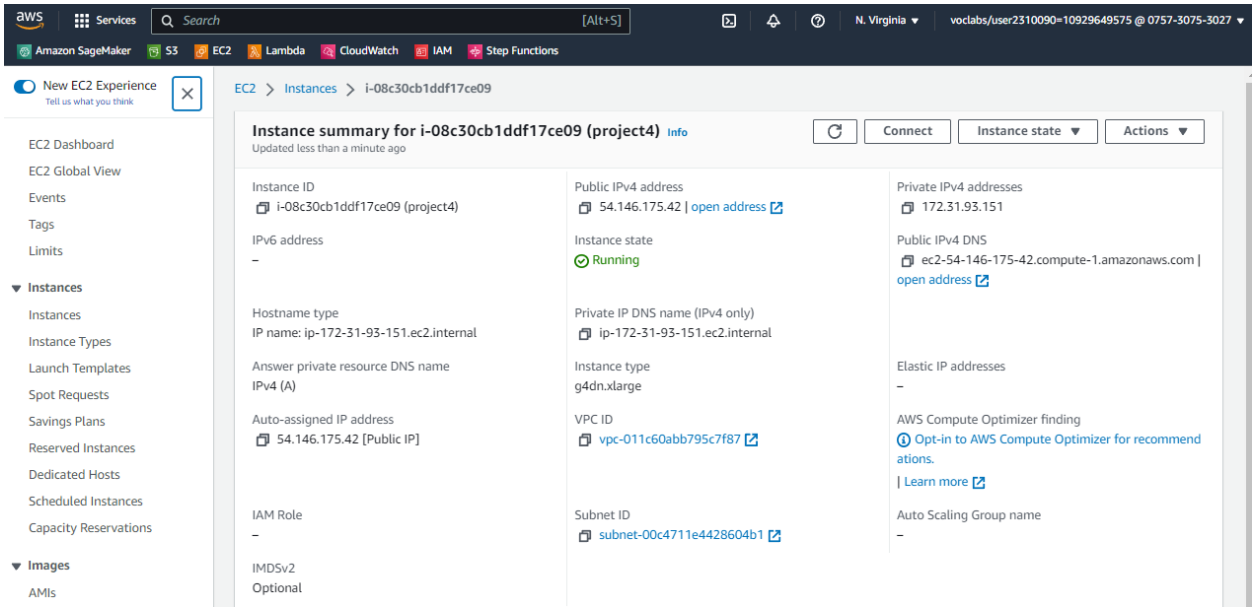


- Endpoints for single instance and multiple instance training are created.

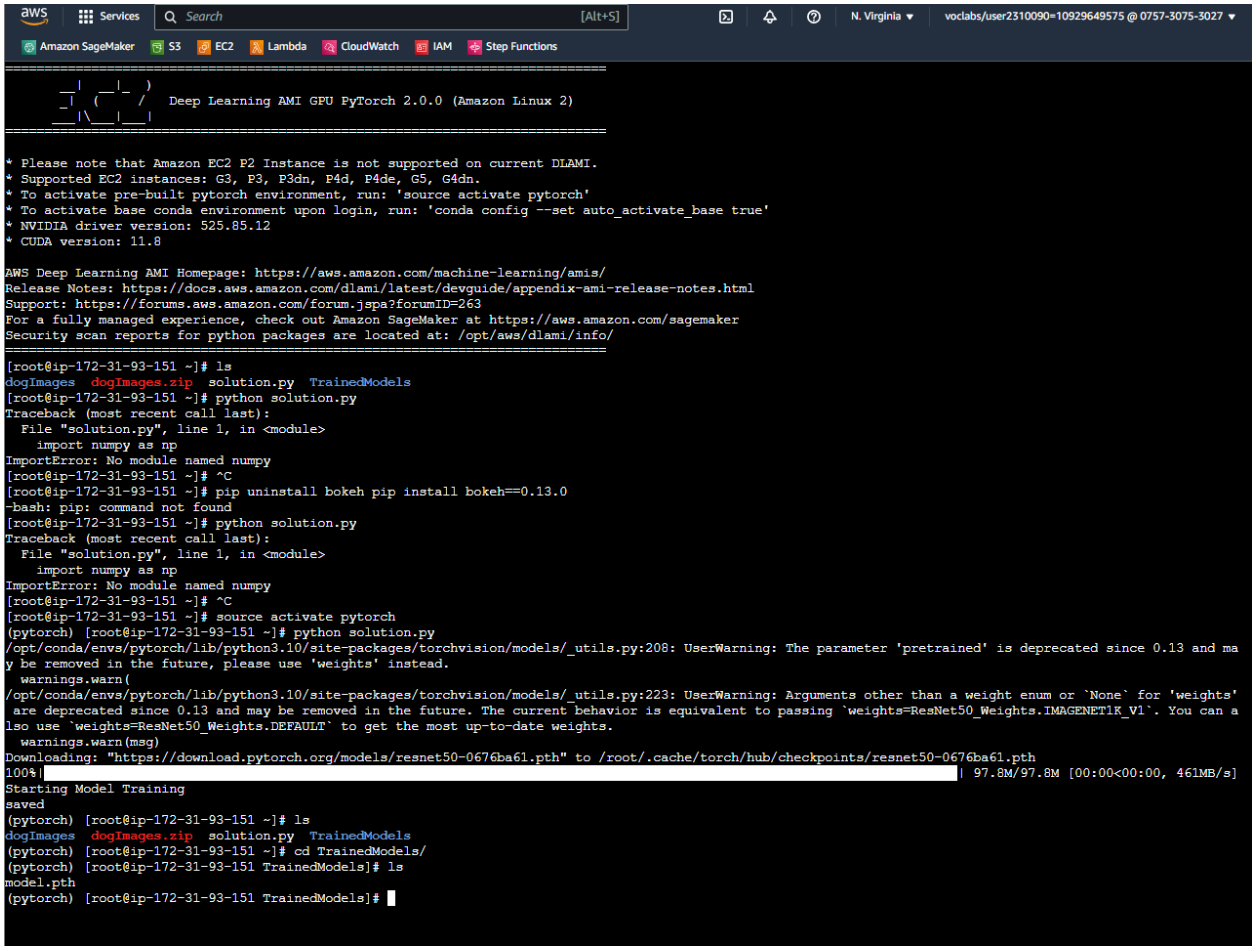


Step 2: Training and saving on EC2

- I have chosen an EC2 instance of **ml.g4dn.xlarge,** tried to choose a larger one but my role did not allow that. System type is Deep Learning AMI GPU PyTorch 2.0.0 since our model is an image classifier.



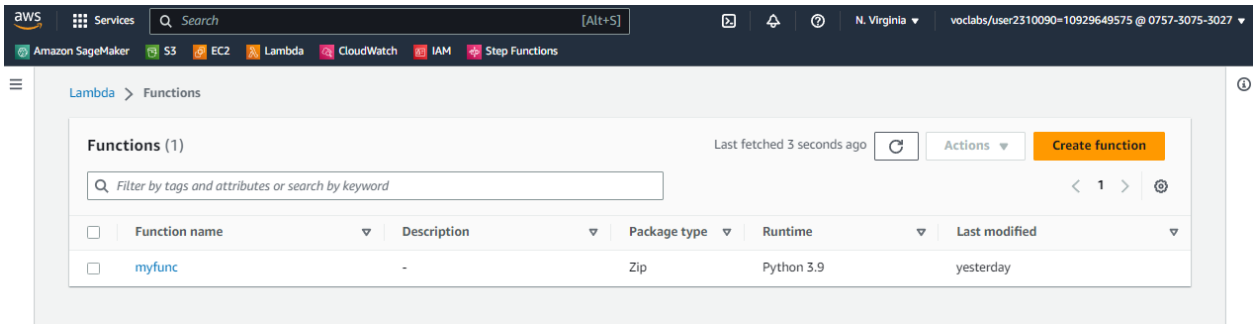
- Successfully created the file and ran. Model is created successfully in TrainedModels. Basically both scripts (step1 and step2) do the same thing. Only difference is that in step 1 we used some Sagemaker libraries to deploy an endpoint for the model we created. This model is stored in s3 like the model stored in EC2 TrianedModels folder. So we use this model to create an endpoint and get predictions.

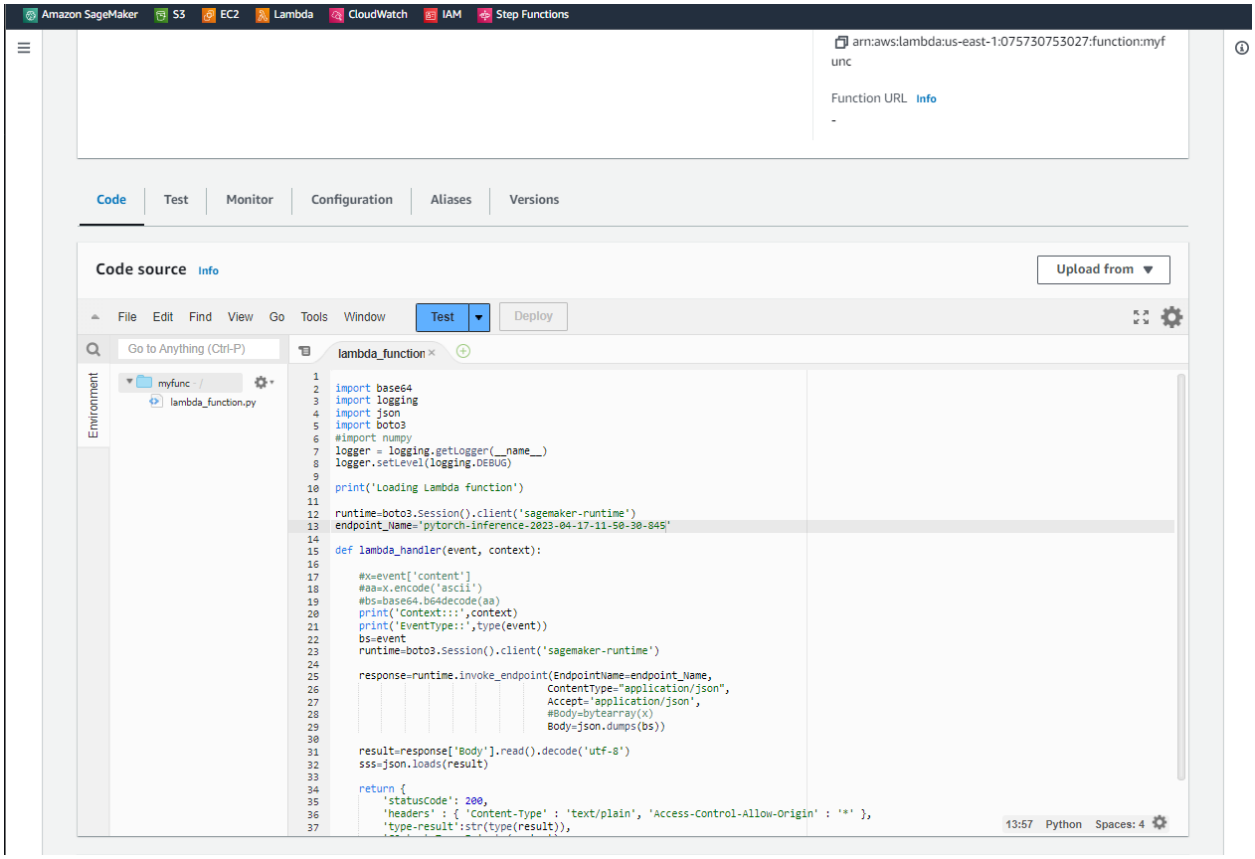


Step 3: Lambda function setup

- A Lambda function named “myfunc” is created and code from source is pasted.
- Endpoint point name is changed

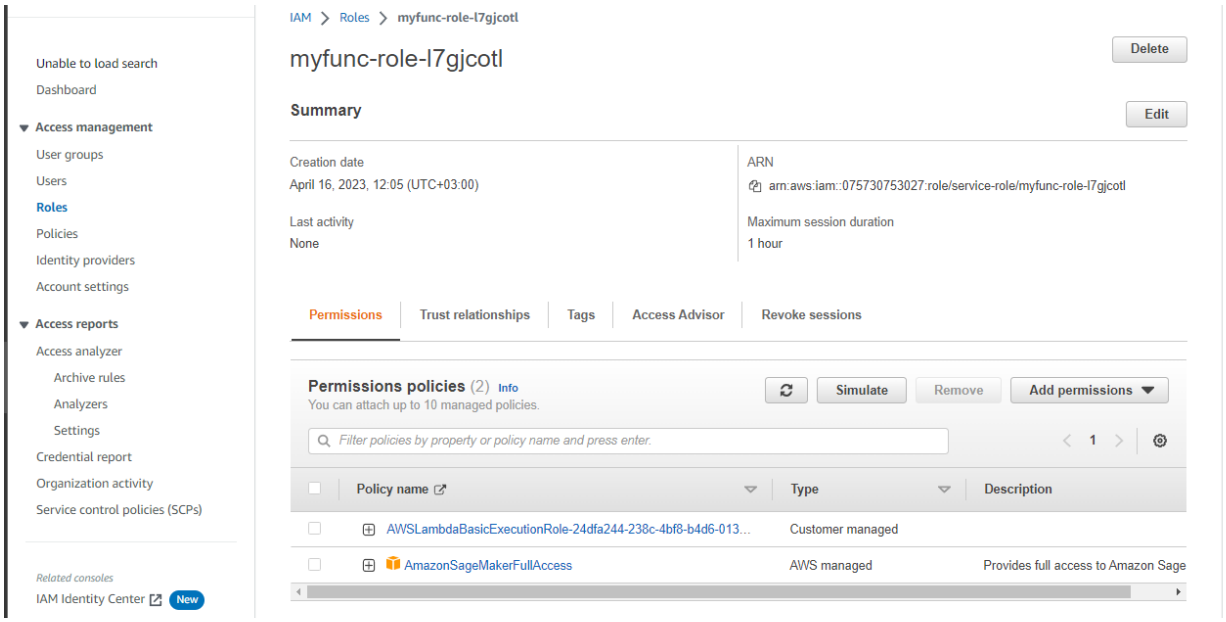
Here the Lambda function we created invokes the endpoint that we created on Step 1. We pass an image url to test the endpoint; if we get success from the request, an array of probabilities is returned, scoring for each of the types of dogs given.





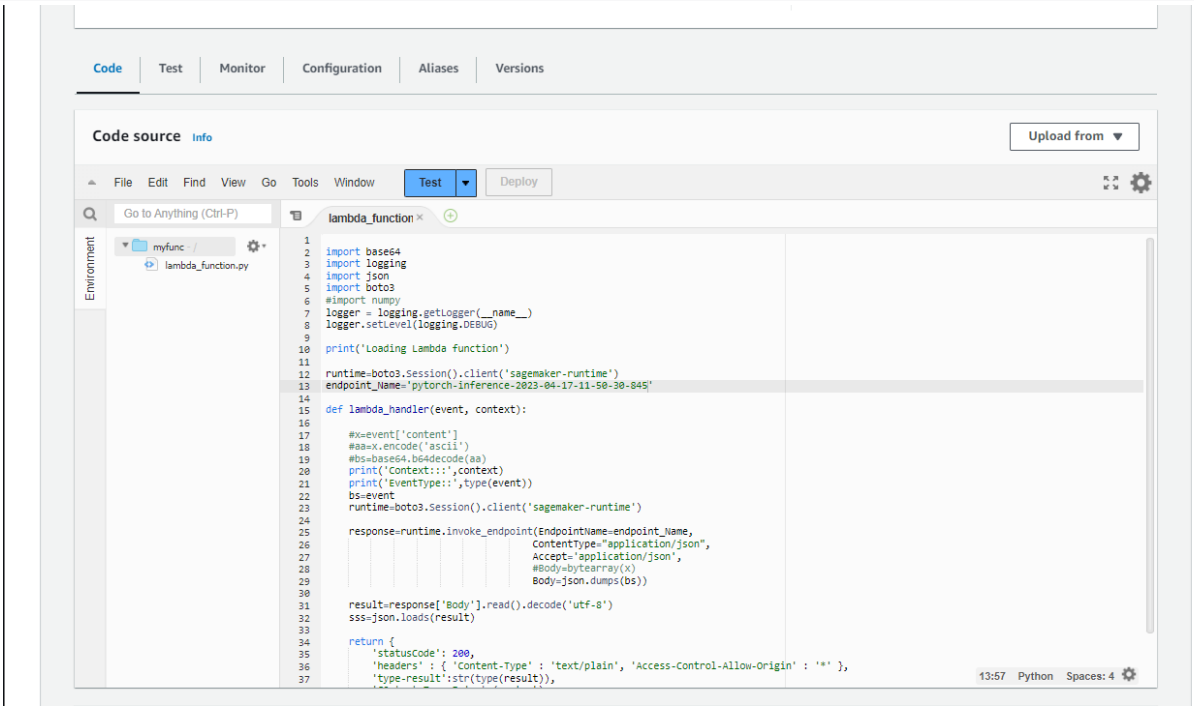
Step 4: Security and testing

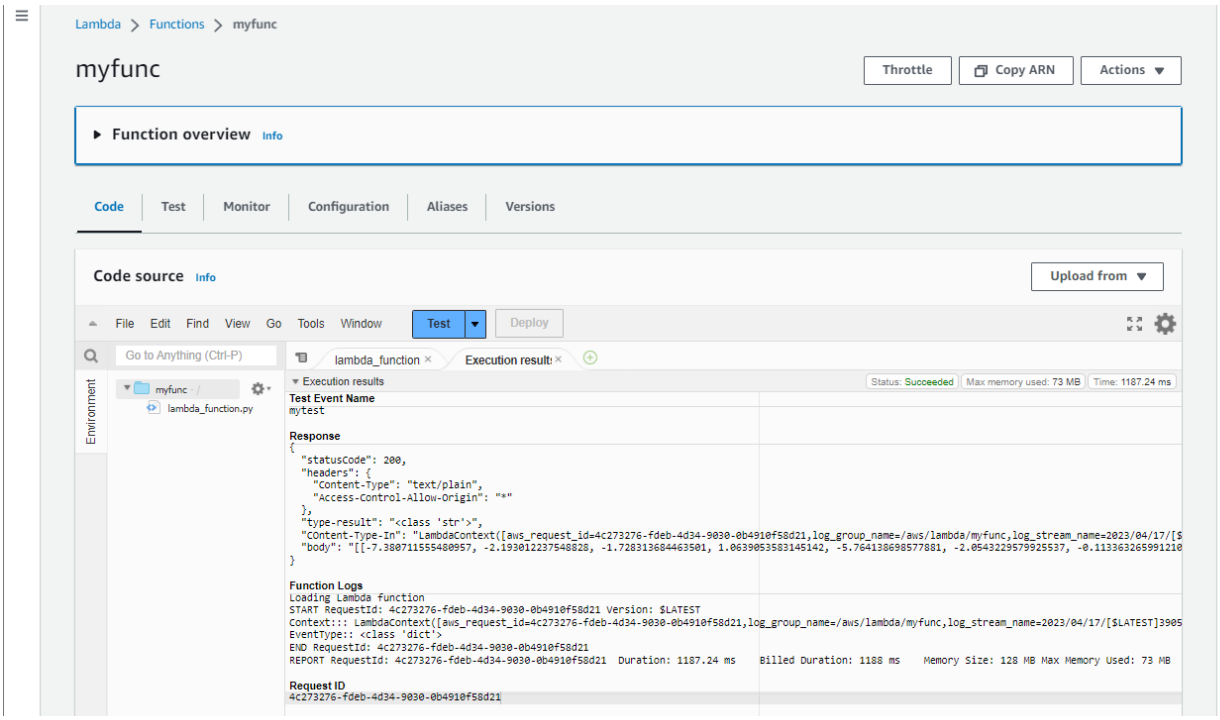
Sagemaker Full Access role is given to the lambda role.



- Lambda function is tested for the given test url:

```
{"url":"https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2017/11/20113314/Carolina-Dog-standing-outdoors.jpg"}
```





- When creating a new Lambda function we are asked whether to use an existing role or create a new one. In this way many roles could be created and we might come across a ton of roles in IAM, also there is a possibility that we could add more policies to these roles. We should always monitor IAM roles and revoke the ones which are not used.

Step 5: Concurrency and auto-scaling

- If we are expecting high traffic we should consider using Concurrency and Autoscaling. For our Lambda function I have both to show up. I have set up a provisioned concurrency with 3 executions and autoscaling up to 3 instances for this lambda function. We should consider the traffic we expect and configure these to avoid bottlenecks.