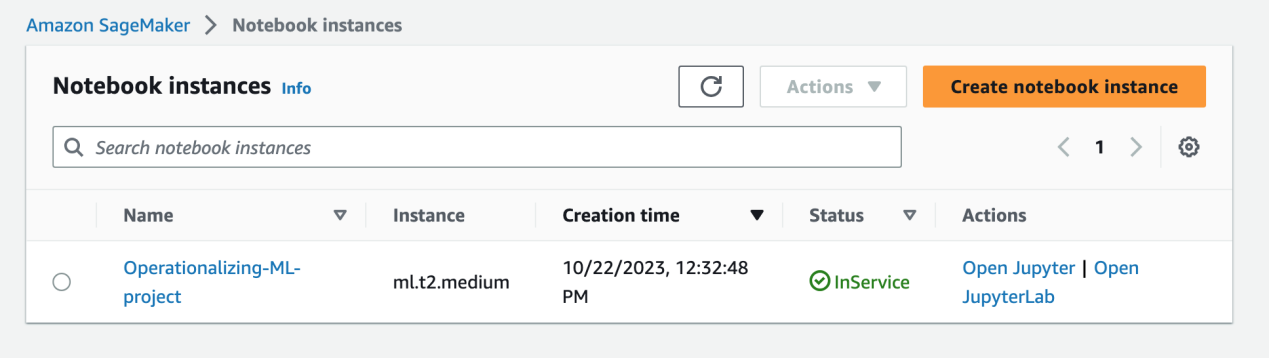
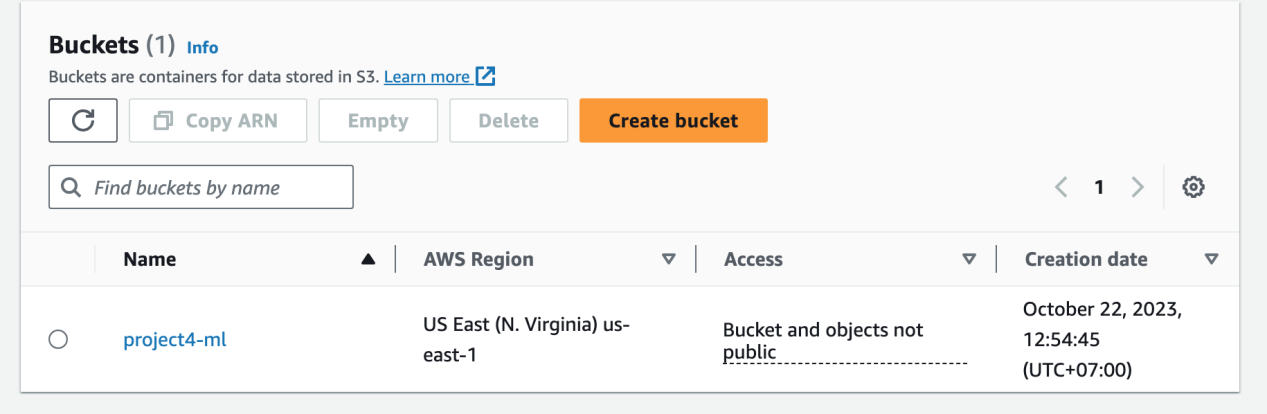
**Step 1: Training and deployment on Sagemaker**

1. **Create a Sagemaker instance**

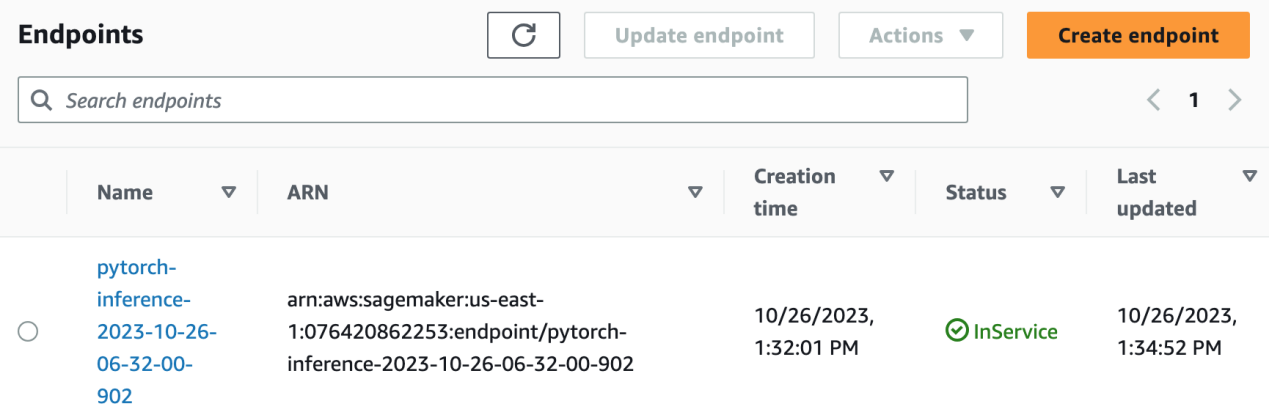
Instance: ml.t2.medium. I chose this instance type due to low-cost.



1. **Create a S3 bucket**

****

1. **Deploy an endpoint**

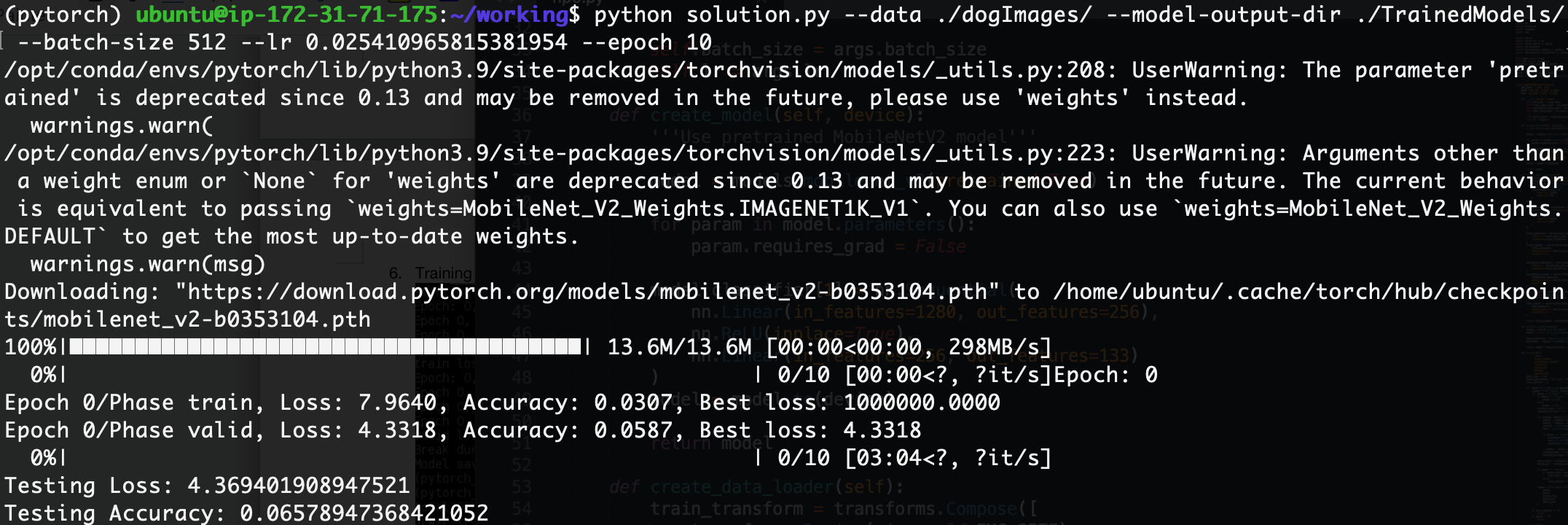
****

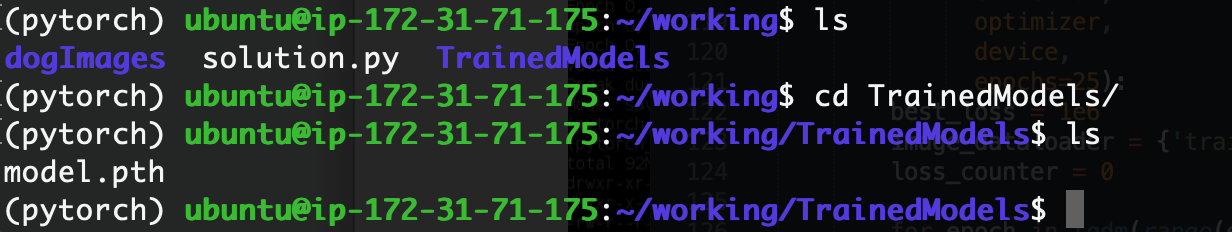
**Step 2: EC2 Training**

1. **EC2 setup**

I chose g4dn.xlarge instance to launch EC2 because I wanted to leverage its GPU capabilities.

1. **Training and saving on EC2**

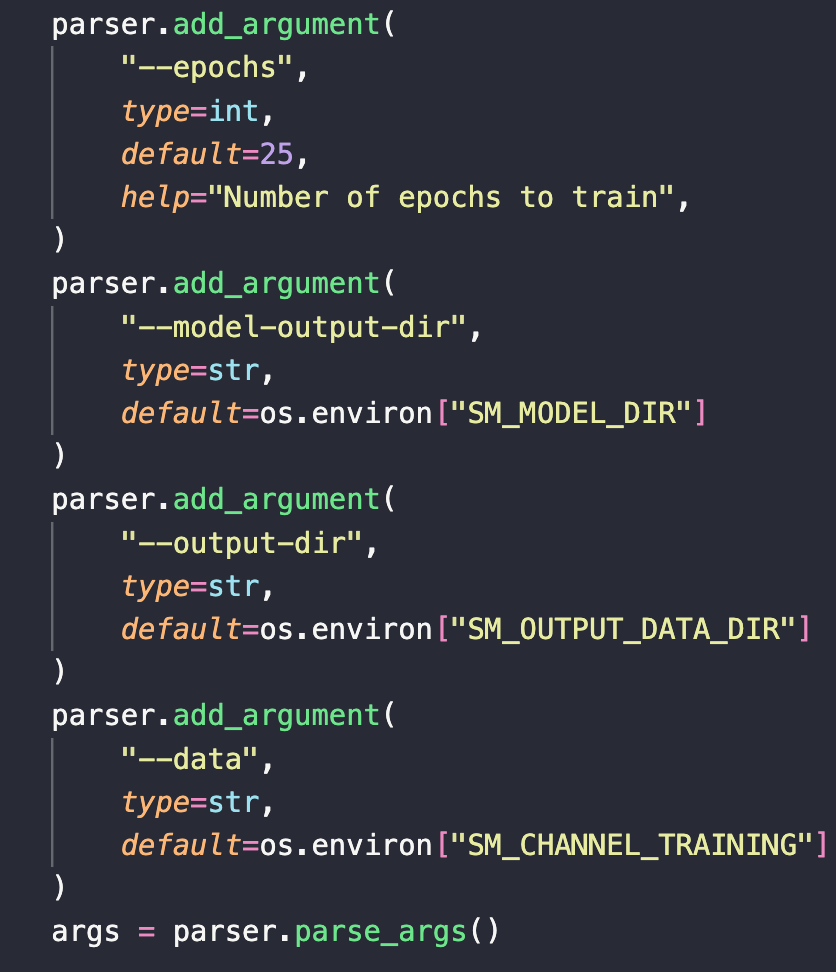




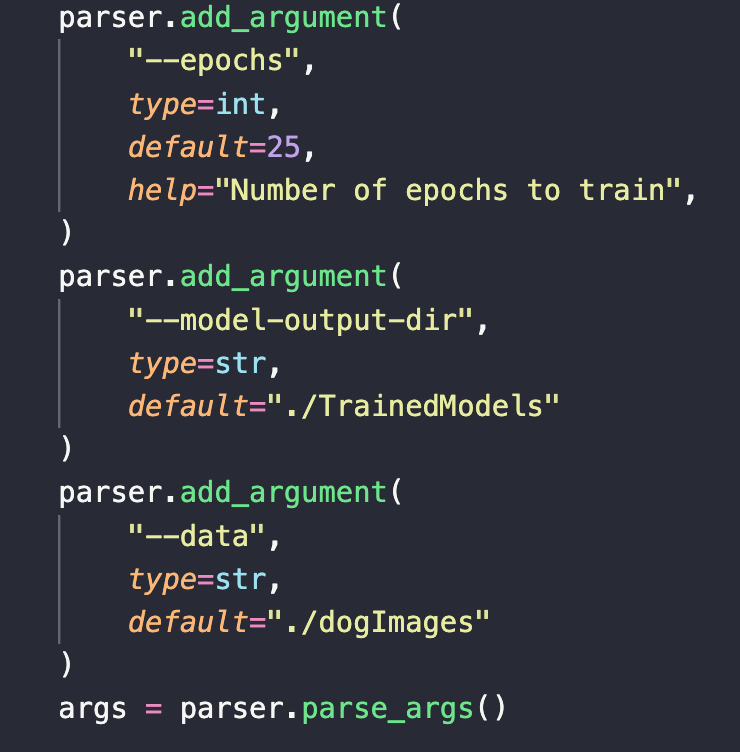
1. **EC2 code**

The script ec2train1.py to use in EC2 is a modification of the script h20.py. I changed data location and model save path parameter.

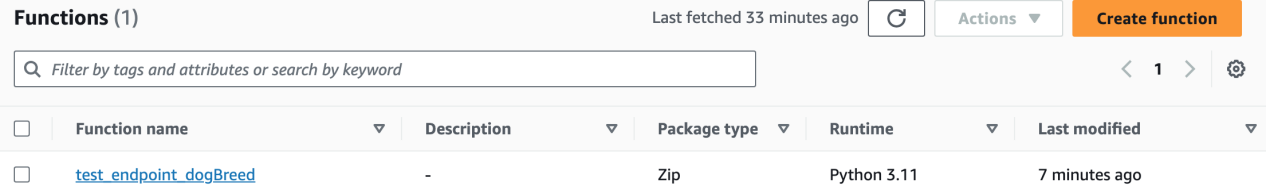
Comparision:

hpo.py:

ec2train1.py



**Step 3: Lambda function setup**

****

**How does lambda\_handler function in lambdafunction.py work?**

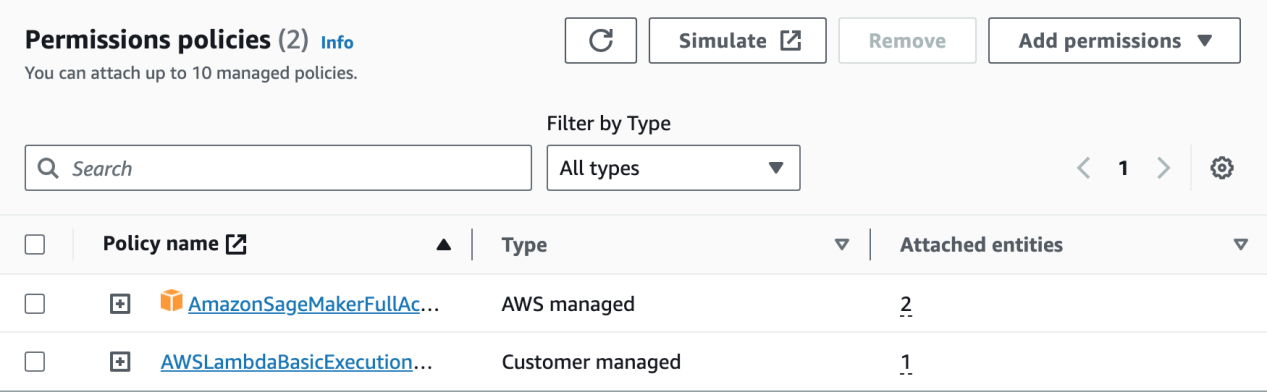
The lambda\_handler function is defined to handle incoming events and context.

Inside lambda\_handler function:

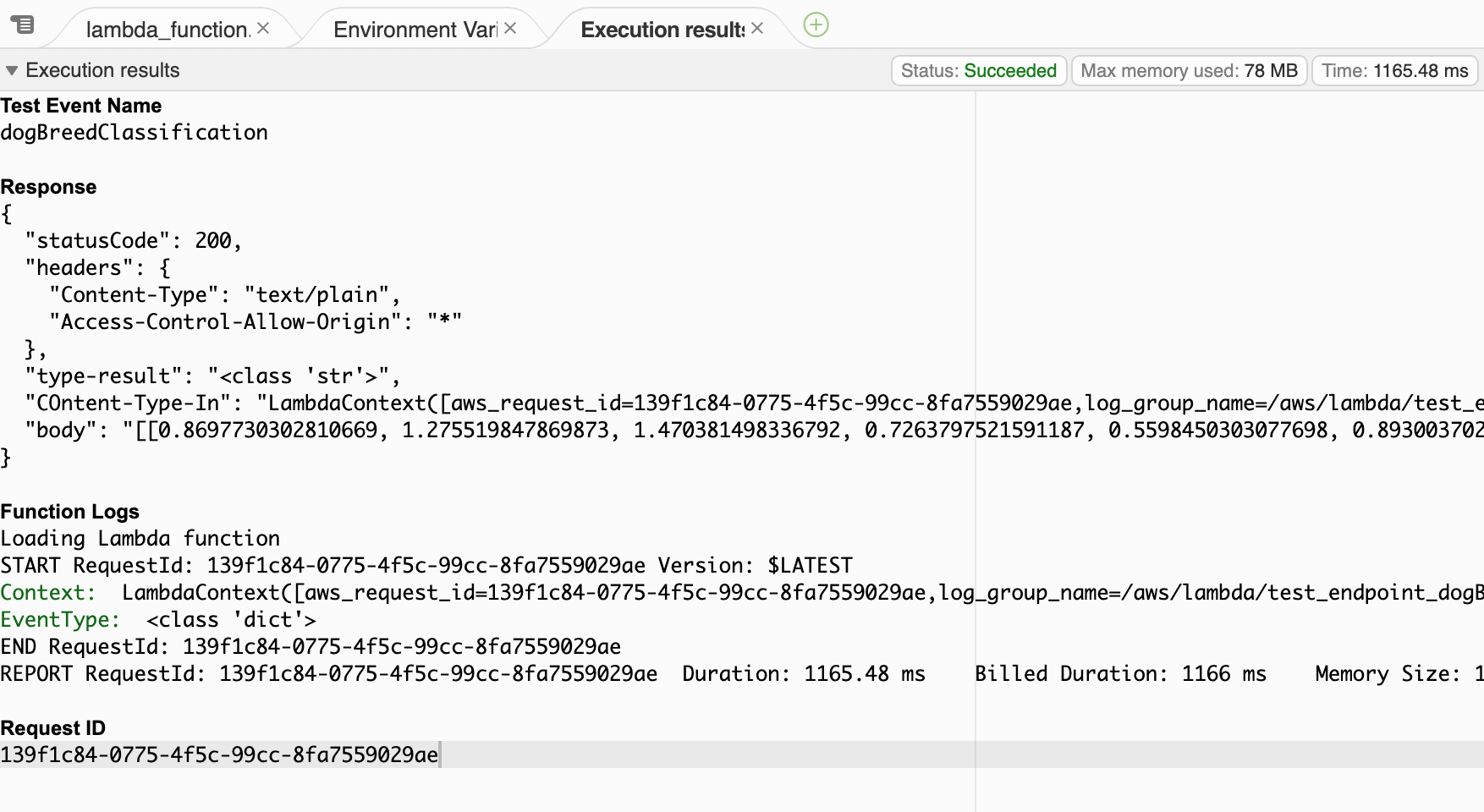
* It invokes the SageMaker endpoint specified by endpoint\_Name. It sends a JSON request with the content type set to "application/json" and expects a JSON response in return.
* It reads and decodes the response from the endpoint and assigns it to the result variable.
* It parses the result as JSON and assigns it to the sss variable.

**Step 4: Security and testing**

We need to setup more permissions for Lambda function:



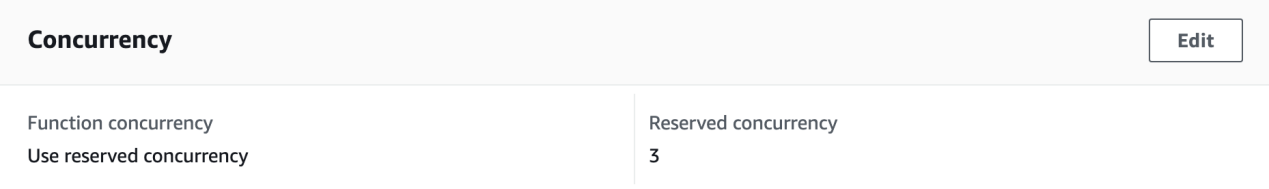
Testing result:

[[0.8697730302810669, 1.275519847869873, 1.470381498336792, 0.7263797521591187, 0.5598450303077698, 0.8930037021636963, 2.5189530849456787, 3.1838414669036865, -0.28565865755081177, 1.8510119915008545, -0.1190861314535141, 1.24089777469635, 1.6205960512161255, 3.9171690940856934, 3.21449613571167, 3.144882917404175, 0.44361263513565063, 1.8896055221557617, 1.2973790168762207, 3.331754684448242, 1.9368903636932373, 1.6240811347961426, -1.35102117061615, -1.5658329725265503, 2.2699174880981445, 1.1281962394714355, -0.6966966986656189, 1.454425573348999, 2.0475924015045166, 1.5677125453948975, 0.9497787356376648, 2.821842670440674, 0.3032481074333191, -0.18353262543678284, 1.117949366569519, 0.38197061419487, 1.6235748529434204, 1.1049245595932007, 2.7361974716186523, 1.8626235723495483, 2.911390781402588, 0.7375891804695129, 3.1136507987976074, 2.321830987930298, 2.4828531742095947, 1.2153549194335938, 2.356417417526245, 2.0313756465911865, 2.2790682315826416, 2.7659409046173096, 0.8223375678062439, 1.2364784479141235, 0.45017266273498535, 0.943295955657959, 1.0359652042388916, 1.9425318241119385, 2.585744857788086, -0.16599197685718536, 2.101928234100342, -1.081555724143982, 1.6206789016723633, 0.6367202401161194, 0.9382332563400269, 0.7766559720039368, 1.481938123703003, 0.1725568324327469, 0.9762061834335327, 0.8009390830993652, 3.0403263568878174, 2.9156455993652344, 1.6722521781921387, 1.6993438005447388, 0.24061641097068787, 1.4696214199066162, 0.04755827784538269, 1.8318250179290771, 0.22392664849758148, 2.765268087387085, 0.7512297630310059, 1.846397876739502, 4.042788028717041, 1.019696593284607, 2.1738226413726807, 0.1482560783624649, 0.5583453178405762, 1.2126249074935913, -0.9322324395179749, 0.7092233896255493, 1.0910390615463257, 3.215555429458618, 0.4462537169456482, 1.4585700035095215, 0.373834490776062, 0.0246770977973938, 0.9141597747802734, 2.6912474632263184, 0.6346117258071899, 0.4494556188583374, 0.9127184152603149, 0.28657862544059753, -0.2256682813167572, 2.0466508865356445, -0.5954168438911438, 0.29967379570007324, 1.8936665058135986, 1.1930088996887207, 1.145112156867981, 0.17924533784389496, 0.9716852903366089, 1.5275604724884033, 0.18016786873340607, 0.5953990817070007, -0.006502941250801086, 0.2940976321697235, -1.5584279298782349, 2.3099377155303955, 0.24616023898124695, 2.6096086502075195, 0.4551774263381958, 3.084825277328491, 1.9219110012054443, 1.608817458152771, 1.2091424465179443, 1.5991531610488892, 0.6568233966827393, 0.2434116154909134, -0.02791639417409897, 1.8432778120040894, 0.9871823787689209, -0.2732100188732147, 0.2898997962474823, 2.6483938694000244, 0.24919497966766357]]

My Lambda function has been granted full access to SageMaker. This is overly permissive and poses a significant security risk. It's crucial to follow the principle of least privilege and grant Lambda function only the specific permissions it needs to perform its tasks. Providing full access can lead to misuse if the function is compromised.

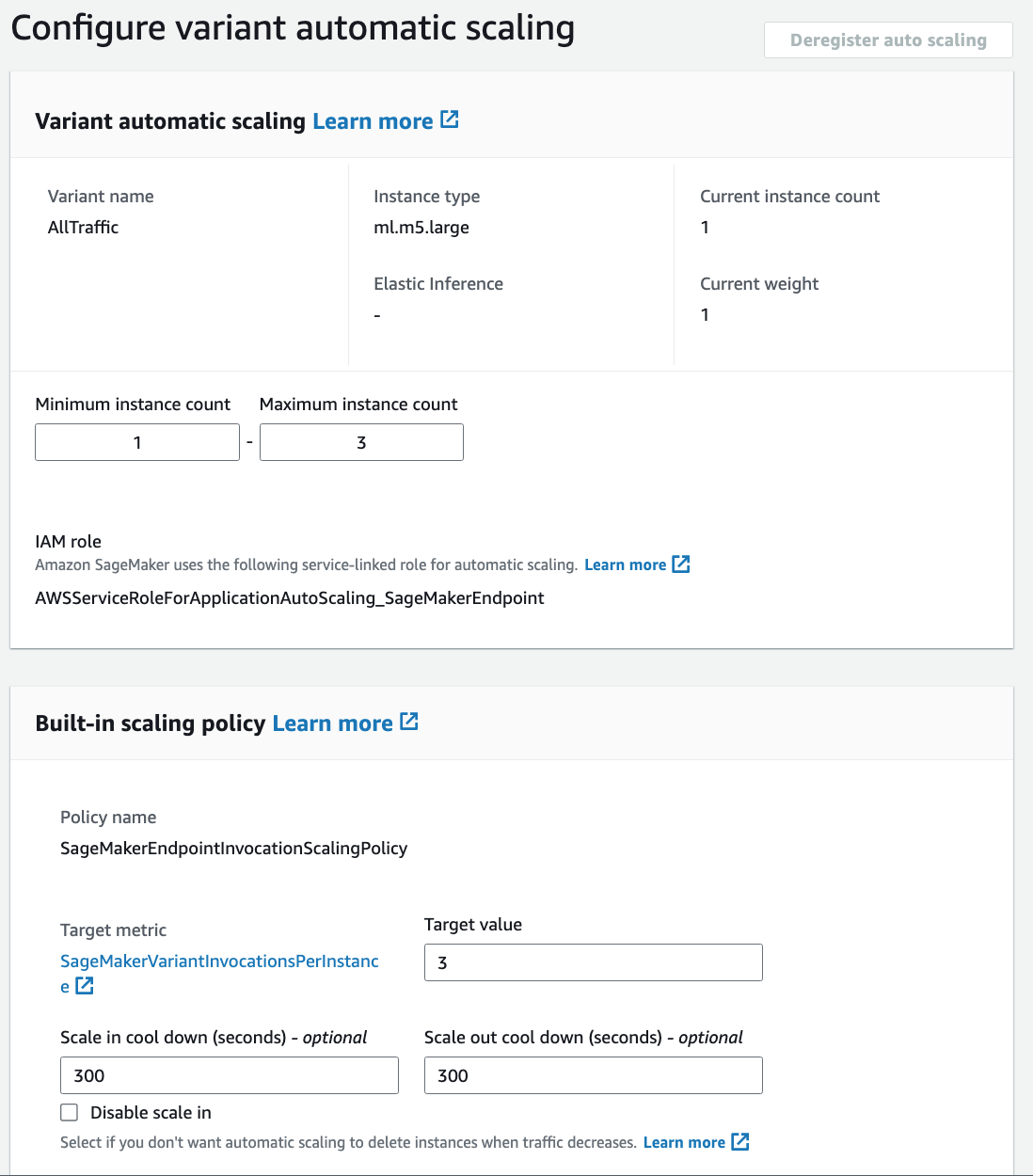
**Step 5: Concurencry and auto-scaling**

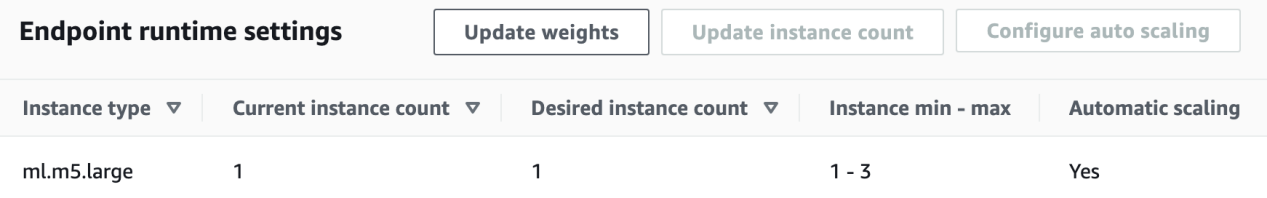
1. **Concurrency**

****

I setup 3 reserved concurrency for my Lambda function to deal with high traffic.

1. **Auto-scaling**

****

****

The maximum instance to scale is 3.

To achieve the highest efficiency, both concurrency and auto-scaling should be used.