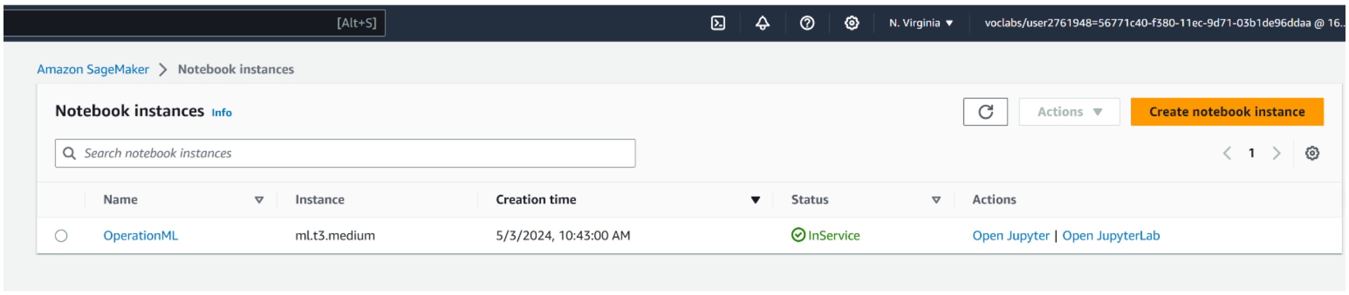
Operationalizing an AWS ML Project

- Ting Lu

1. Initial Setup

I have chosen the "**ml.t2.medium**" instance type for Notebook instance (Figure 1 - SageMaker Notebook Instance). There are multiple reasons for selecting this instance type for my notebook.

- Firstly, for completing the execution of this project's jupyter notebooks we **do not need** a very computationally **powerful CPU** and **high RAM.**
- We will need to keep this **notebook** instance in "**inService**" **status** for a **long time** while we are working on the project
- In order to **avoid high costs,** we should **select a notebook** that is **low in per hour cost** and **offers** reasonably **good CPU and RAM**.
- So looking at the instance type and their pricing: https://aws.amazon.com/SageMaker/pricing/ , we have two choices "ml.t2.medium" and "ml.t3.medium". Both have 2 vCPU and 4 GB Memory, and as per doc "ml.t3.medium" has a slightly higher cost as it has a fast boot time.
- Now given that we do have a critical requirement for a fast boot time, so we can go ahead with the "**ml.t2.medium**" as it offers the same computational power and is lower in per hour cost.



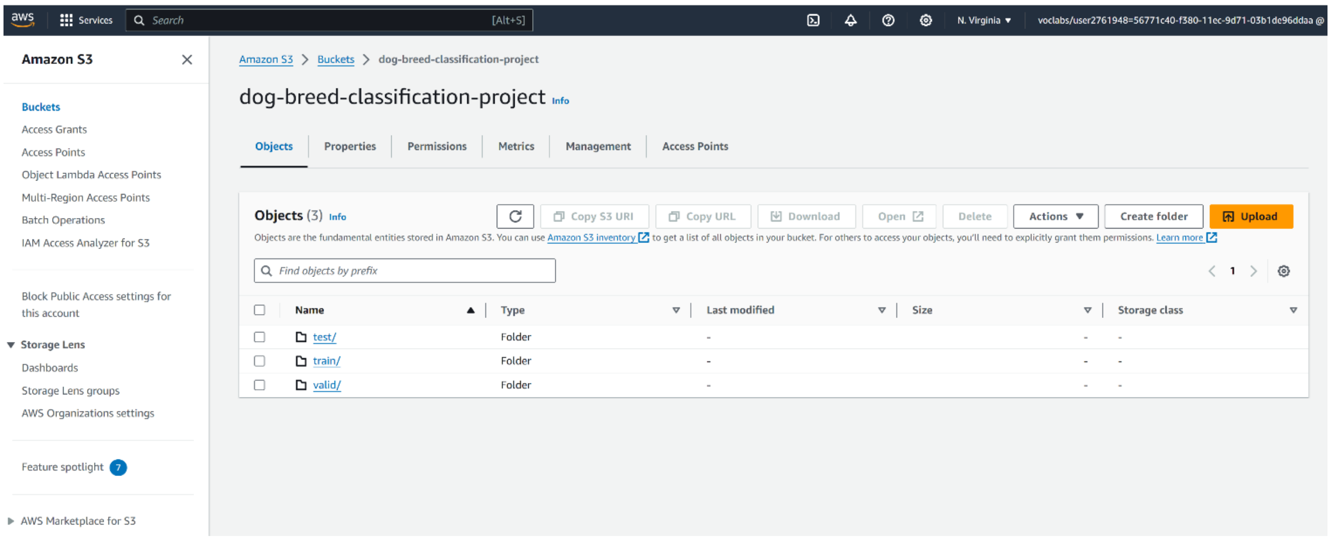
Amazon SageMaker > Notebook instances

Notebook instances Info

Name	Instance	Creation time	Status	Actions
OperationML	ml.t3.medium	5/3/2024, 10:43:00 AM	InService	Open Jupyter | Open JupyterLab

Figure 1. SageMaker Notebook Instance

The dog breed dataset was uploaded to a newly created S3 bucket, successfully.



Amazon S3 > Buckets > dog-breed-classification-project

dog-breed-classification-project Info

Objects (3) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

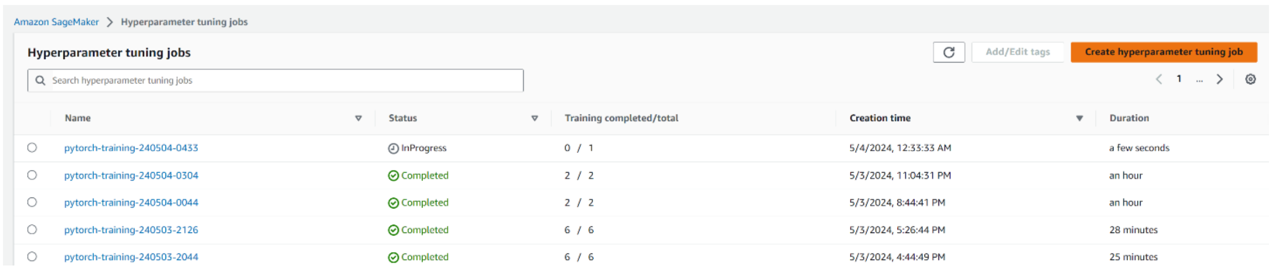
Name	Type	Last modified	Size	Storage class
test/	Folder	-	-	-
train/	Folder	-	-	-
valid/	Folder	-	-	-

Figure 2. S3 Bucket snapshot

2. SageMaker Training and Deployment

For hyperparameter tuning I used the same “ml.m5.xlarge” instance_type , however since it took too much time for the tuning, while the training process I tried to increase the processing power a bit by using the “ml.m5.2xlarge” for the single instance and multi instance training purposes.

Hyperparameter tuning job (max_jobs = 2, max_parallel_jobs = 2) completed successfully:


Amazon SageMaker > Hyperparameter tuning jobs

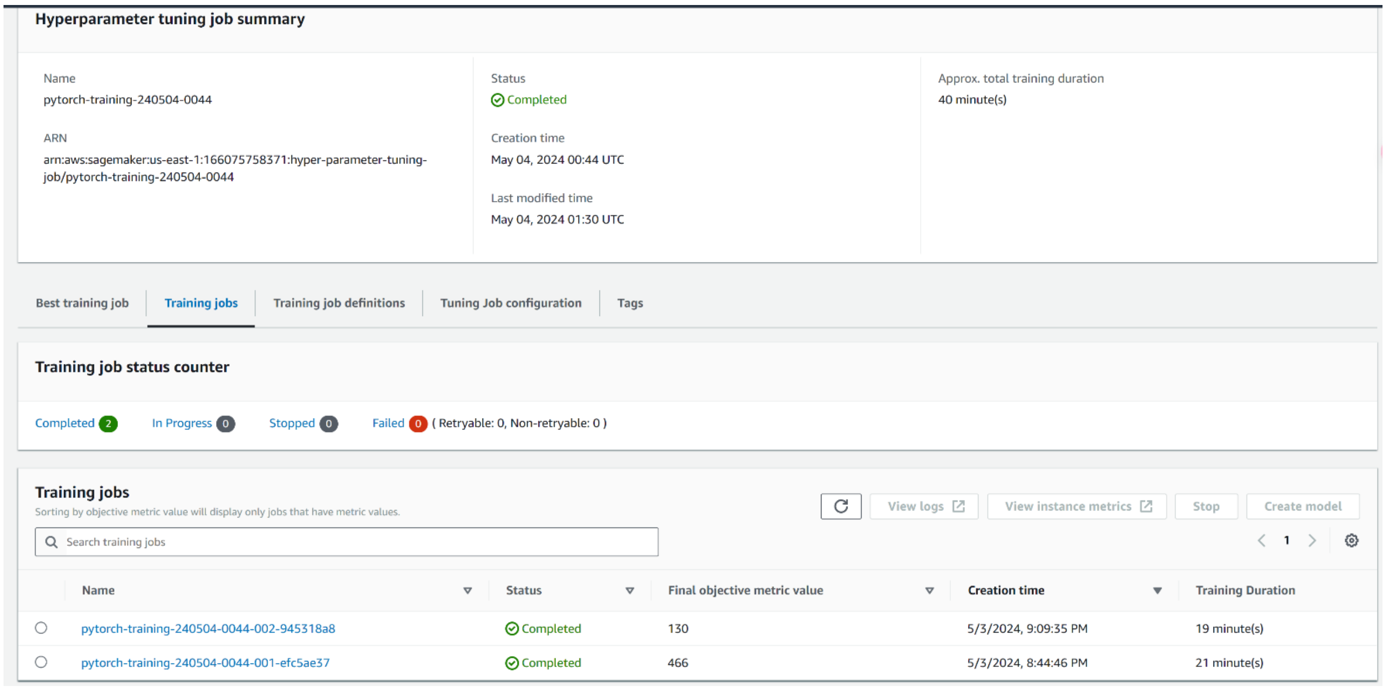
Hyperparameter tuning jobs

	Name	Status	Training completed/total	Creation time	Duration
○	pytorch-training-240504-0433	InProgress	0 / 1	5/4/2024, 12:33:33 AM	a few seconds
○	pytorch-training-240504-0304	Completed	2 / 2	5/3/2024, 11:04:31 PM	an hour
○	pytorch-training-240504-0044	Completed	2 / 2	5/3/2024, 8:44:41 PM	an hour
○	pytorch-training-240503-2126	Completed	6 / 6	5/3/2024, 5:26:44 PM	28 minutes
○	pytorch-training-240503-2044	Completed	6 / 6	5/3/2024, 4:44:49 PM	25 minutes

Figure 3. Hyperparameter Tuning Job

However, upon training the model with the best parameters from above tuning, the model gave a 0 test accuracy! So increased the max_jobs = 6, max_parallel_jobs = 3 and also changed its instance_type = “ml.m5.xlarge” to speed up the computations a bit.

Reran the hyperparameter jobs and it executed successfully:


Hyperparameter tuning job summary

Name
pytorch-training-240504-0044

ARN
arn:aws:sagemaker:us-east-1:166075758371:hyper-parameter-tuning-job/pytorch-training-240504-0044

Status
Completed

Creation time
May 04, 2024 00:44 UTC

Last modified time
May 04, 2024 01:30 UTC

Approx. total training duration
40 minute(s)

Training job status counter

Completed 2 In Progress 0 Stopped 0 Failed 0 (Retryable: 0, Non-retryable: 0)

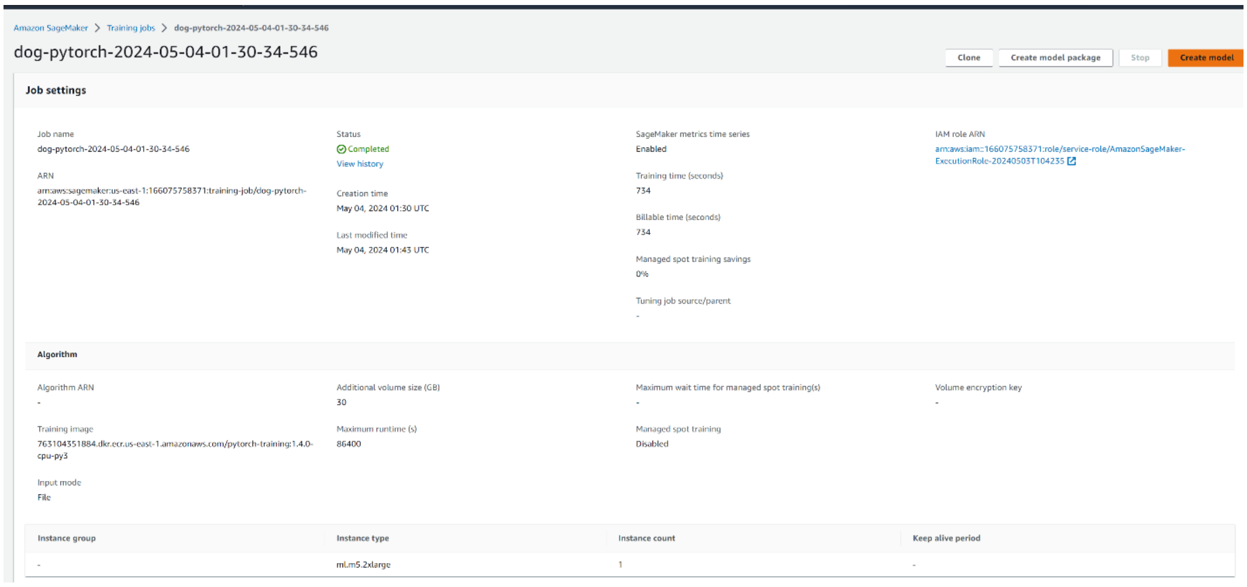
Training jobs

Sorting by objective metric value will display only jobs that have metric values.

	Name	Status	Final objective metric value	Creation time	Training Duration
○	pytorch-training-240504-0044-002-945318a8	Completed	130	5/3/2024, 9:09:35 PM	19 minute(s)
○	pytorch-training-240504-0044-001-efc5ae37	Completed	466	5/3/2024, 8:44:46 PM	21 minute(s)

Figure 4. Hyperparameter jobs summary

Post which triggered the **single instance** and **multi-instance training jobs**. Jobs completed successfully. For snapshot of the training jobs refer the images folder in the repository.



Amazon SageMaker > Training jobs > dog-pytorch-2024-05-04-01-30-34-546
Clone
Create model package
Stop
Create model

dog-pytorch-2024-05-04-01-30-34-546

Job settings

Job name
dog-pytorch-2024-05-04-01-30-34-546

ARN
arn:aws:sagemaker:us-east-1:166075758371:training-job/dog-pytorch-2024-05-04-01-30-34-546

Status
Completed
View history

Creation time
May 04, 2024 01:30 UTC

Last modified time
May 04, 2024 01:43 UTC

SageMaker metrics time series
Enabled

Training time (seconds)
734

Billable time (seconds)
734

Managed spot training savings
0%

Tuning job source/parent
-

IAM role ARN
arn:aws:iam::166075758371:role/service-role/AmazonSageMaker-ExecutionRole-20240503T104235

Algorithm

Algorithm ARN
-

Training image
763104351884.dkr.ecr.us-east-1.amazonaws.com/pytorch-training:1.4.0-cpu-py3

Input mode
File

Additional volume size (GB)
30

Maximum runtime (s)
86400

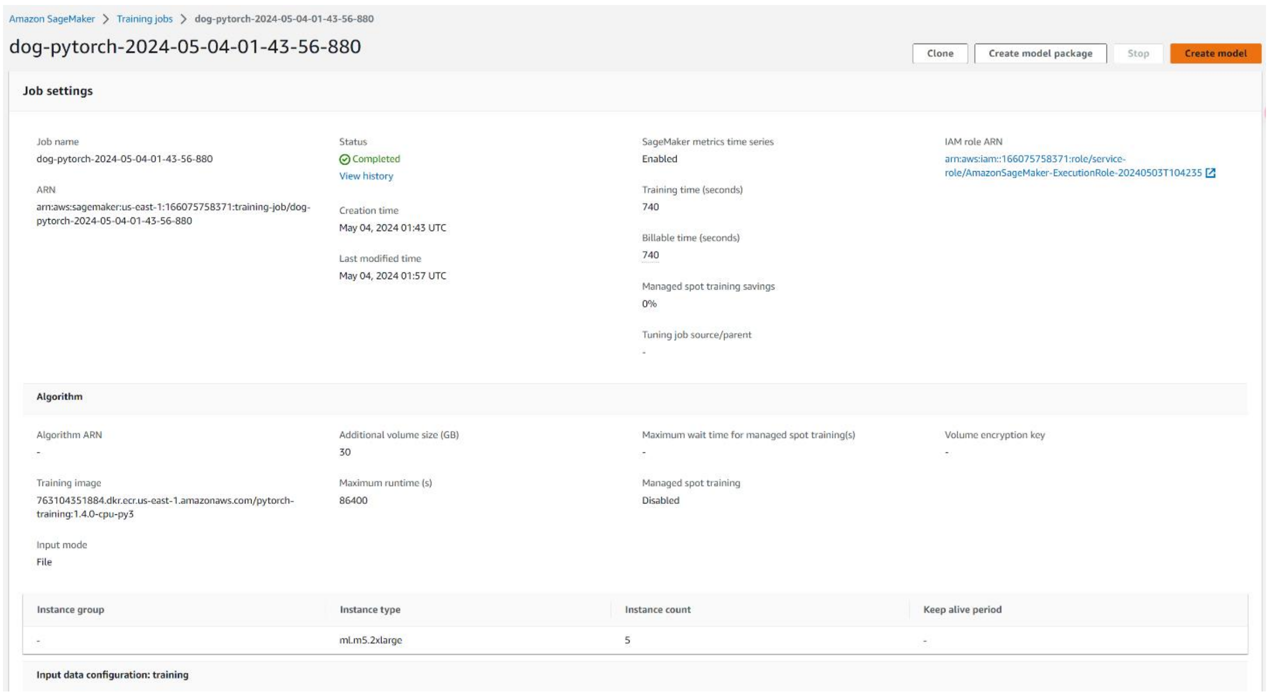
Maximum wait time for managed spot training(s)
-

Managed spot training
Disabled

Volume encryption key
-

Instance group	Instance type	Instance count	Keep alive period
-	ml.m5.2xlarge	1	-

Figure 5. Single Instance Training Job



Amazon SageMaker > Training jobs > dog-pytorch-2024-05-04-01-43-56-880
Clone
Create model package
Stop
Create model

dog-pytorch-2024-05-04-01-43-56-880

Job settings

Job name
dog-pytorch-2024-05-04-01-43-56-880

ARN
arn:aws:sagemaker:us-east-1:166075758371:training-job/dog-pytorch-2024-05-04-01-43-56-880

Status
Completed
View history

Creation time
May 04, 2024 01:43 UTC

Last modified time
May 04, 2024 01:57 UTC

SageMaker metrics time series
Enabled

Training time (seconds)
740

Billable time (seconds)
740

Managed spot training savings
0%

Tuning job source/parent
-

IAM role ARN
arn:aws:iam::166075758371:role/service-role/AmazonSageMaker-ExecutionRole-20240503T104235

Algorithm

Algorithm ARN
-

Training image
763104351884.dkr.ecr.us-east-1.amazonaws.com/pytorch-training:1.4.0-cpu-py3

Input mode
File

Additional volume size (GB)
30

Maximum runtime (s)
86400

Maximum wait time for managed spot training(s)
-

Managed spot training
Disabled

Volume encryption key
-

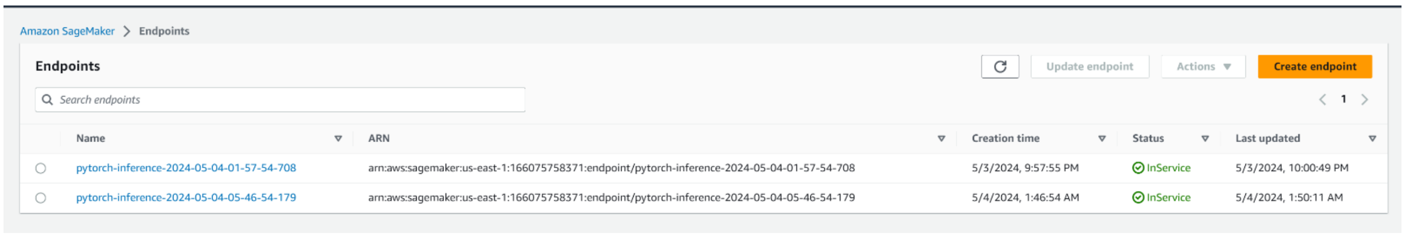
Instance group	Instance type	Instance count	Keep alive period
-	ml.m5.2xlarge	5	-

Input data configuration: training

Figure 6. Multi Instance Training Job

Deployed Endpoints:

- Single instance deployed endpoint: “pytorch-inference-2024-05-04-01-57-54-708”
- Multi instance deployed endpoint: “pytorch-inference-2024-05-04-05-46-54-179”



Endpoints

Name	ARN	Creation time	Status	Last updated
pytorch-inference-2024-05-04-01-57-54-708	arn:aws:sagemaker:us-east-1:166075758371:endpoint/pytorch-inference-2024-05-04-01-57-54-708	5/3/2024, 9:57:55 PM	InService	5/3/2024, 10:00:49 PM
pytorch-inference-2024-05-04-05-46-54-179	arn:aws:sagemaker:us-east-1:166075758371:endpoint/pytorch-inference-2024-05-04-05-46-54-179	5/4/2024, 1:46:54 AM	InService	5/4/2024, 1:50:11 AM

Figure 7. SageMaker Endpoints

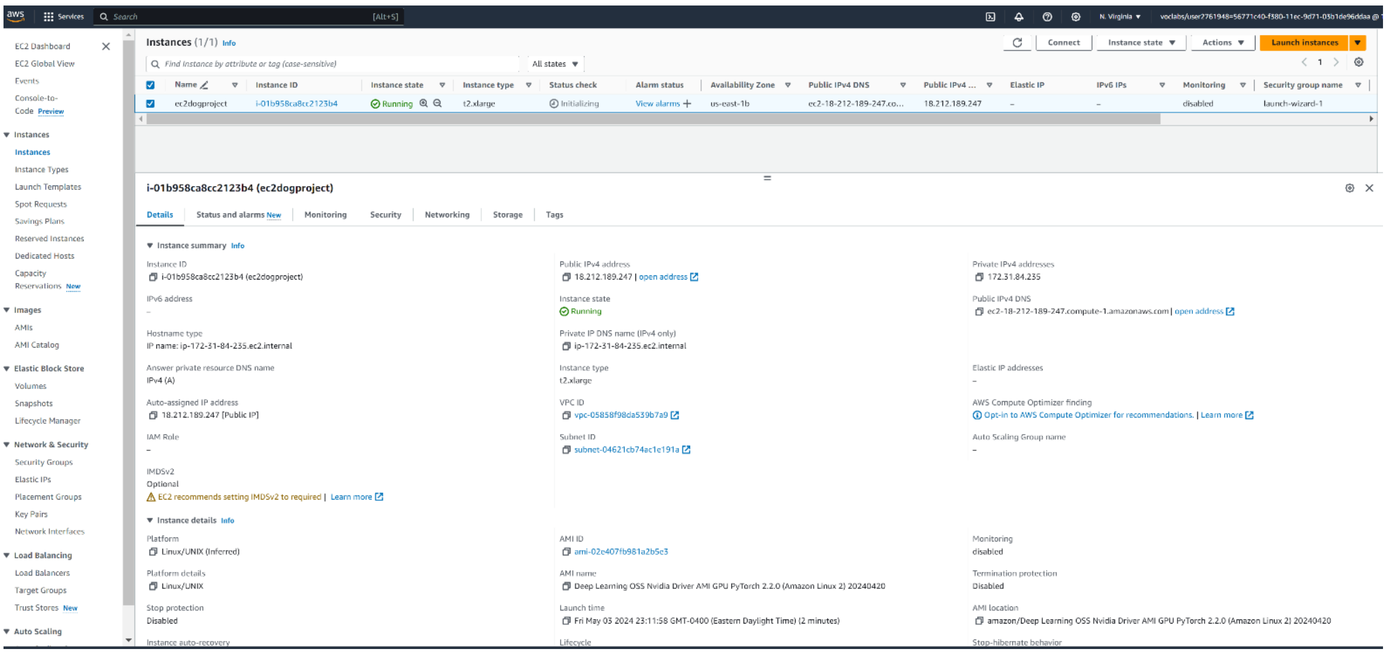
3. EC2 Training

- We have utilized the **t2.xlarge** instance and the **Deep Learning AMI (Amazon Linux 2) Version 55.0**. This seems like a reasonable balance of performance and affordability.
- As per the documentation, T2 instances can sustain high CPU performance for as long as a workload needs it.
- For most general-purpose workloads, T2 instances will provide ample performance without any additional charges.
- Similarly, because we don't know the duration for which we might need to keep this EC2 instance running for training, it's better to go with a medium size instance so we don't have to pay for a large instance while we're doing setup, debugging and other tasks.

Difference between ec2train1.py (EC2 script) and train_and_deploy-solution.ipynb + hpo.py (SageMaker scripts)

- There is no logic for calling any Estimator or Tuner functions in the EC2 script. The code in the EC2 script is responsible for saving the model to the local path. While in the SageMaker scripts this was handled internally by SageMaker where the model data was stored to a S3 location.
- In the EC2 training script, all the variables like hyperparameters and output locations, etc are already mentioned in the script itself and so there is no need for **argparse.** Meaning while running the EC2 script we do not need to mention any arguments.
- In the EC2 script the training happens on the same server on which the script is invoked/executed, however in the SageMaker scripts the training job that is invoked, it runs on a separate container than the one on which the SageMaker notebook is running.
- Another difference is that ec2train1.py lacks the main function
- For the EC2 Training, given that the training data and model, all are stored on the EC2 instance host itself it would be difficult to deploy the saved model to an endpoint in SageMaker. If we wish to do that then we might need to manually upload the model first to SageMaker and then use that to deploy an endpoint. This is not the case in models trained via the SageMaker notebook instances, as the model can be easily deployed to an endpoint.

For snapshot refer below:



Instances (1/1) Info

Find instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name
ec2dogproject	i-01b958ca8cc2123b4	Running	t2.xlarge	Initializing	View alarms +	us-east-1b	ec2-18-212-189-247.co...	18.212.189.247	–	–	disabled	launch-wizard-1

i-01b958ca8cc2123b4 (ec2dogproject)

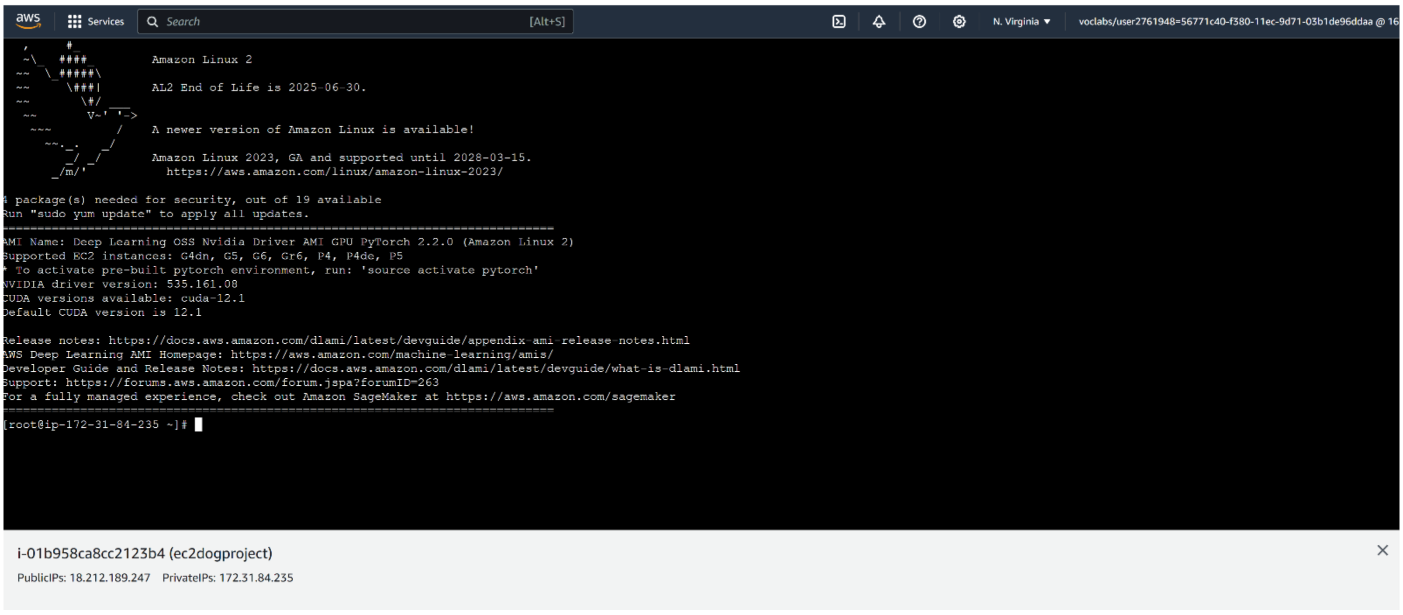
Instance summary Info

Instance ID: i-01b958ca8cc2123b4 (ec2dogproject)	Public IPv4 address: 18.212.189.247 | open address	Private IPv4 addresses: 172.31.84.235
IPv6 address: –	Instance state: Running	Public IPv4 DNS: ec2-18-212-189-247.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-172-31-84-235.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-84-235.ec2.internal	
Answer private resource DNS name: IPv4 (A)	Instance type: t2.xlarge	Elastic IP addresses: –
Auto-assigned IP address: 18.212.189.247 [Public IP]	VPC ID: vpc-05858f98da539b7a9	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –	Subnet ID: subnet-04621cb74ac1e191a	Auto Scaling Group name: –
IMDSv2: Optional — EC2 recommends setting IMDSv2 to required | Learn more		

Instance details Info

Platform: Linux/UNIX (Inferred)	AMI ID: ami-02e407fb981a2b5e3	Monitoring: disabled
Platform details: Linux/UNIX	AMI name: Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.2.0 (Amazon Linux 2) 20240420	Termination protection: Disabled
Stop protection: Disabled	Launch time: Fri May 03 2024 23:11:58 GMT-0400 (Eastern Daylight Time) (2 minutes)	AMI location: amazon/Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.2.0 (Amazon Linux 2) 20240420
Instance auto-recovery	Lifecycle	Stop-hibernate behavior

Figure 8. EC2 Instance snapshot



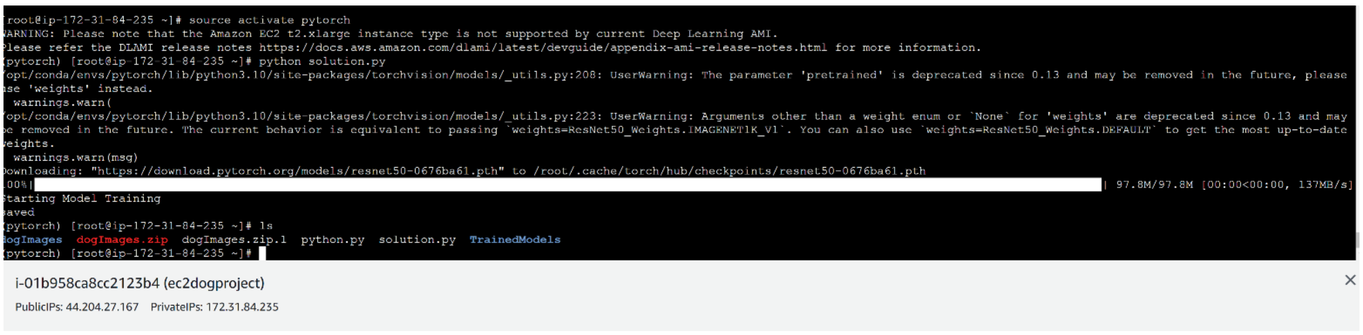
```
  ,     #_
  ~\_  ####_        Amazon Linux 2
 ~~  \_#####\
 ~~     \###|       AL2 End of Life is 2025-06-30.
 ~~       \#/ ___
  ~~       V~' '->
   ~~~         /    A newer version of Amazon Linux is available!
     ~~._.   _/
        _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
      _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

4 package(s) needed for security, out of 19 available
Run "sudo yum update" to apply all updates.
=============================================================================
AMI Name: Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.2.0 (Amazon Linux 2)
Supported EC2 instances: G4dn, G5, G6, Gr6, P4, P4de, P5
* To activate pre-built pytorch environment, run: 'source activate pytorch'
NVIDIA driver version: 535.161.08
CUDA versions available: cuda-12.1
Default CUDA version is 12.1

Release notes: https://docs.aws.amazon.com/dlami/latest/devguide/appendix-ami-release-notes.html
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/amis/
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/devguide/what-is-dlami.html
Support: https://forums.aws.amazon.com/forum.jspa?forumID=263
For a fully managed experience, check out Amazon SageMaker at https://aws.amazon.com/sagemaker
=============================================================================
[root@ip-172-31-84-235 ~]#
```

i-01b958ca8cc2123b4 (ec2dogproject)

PublicIPs: 18.212.189.247 PrivateIPs: 172.31.84.235

Services
Search [Alt+S]
N. Virginia
voclabs/user2761948=56771c40-f380-11ec-9d71-03b1de96ddaa @ 16

Figure 9. EC2 Terminal

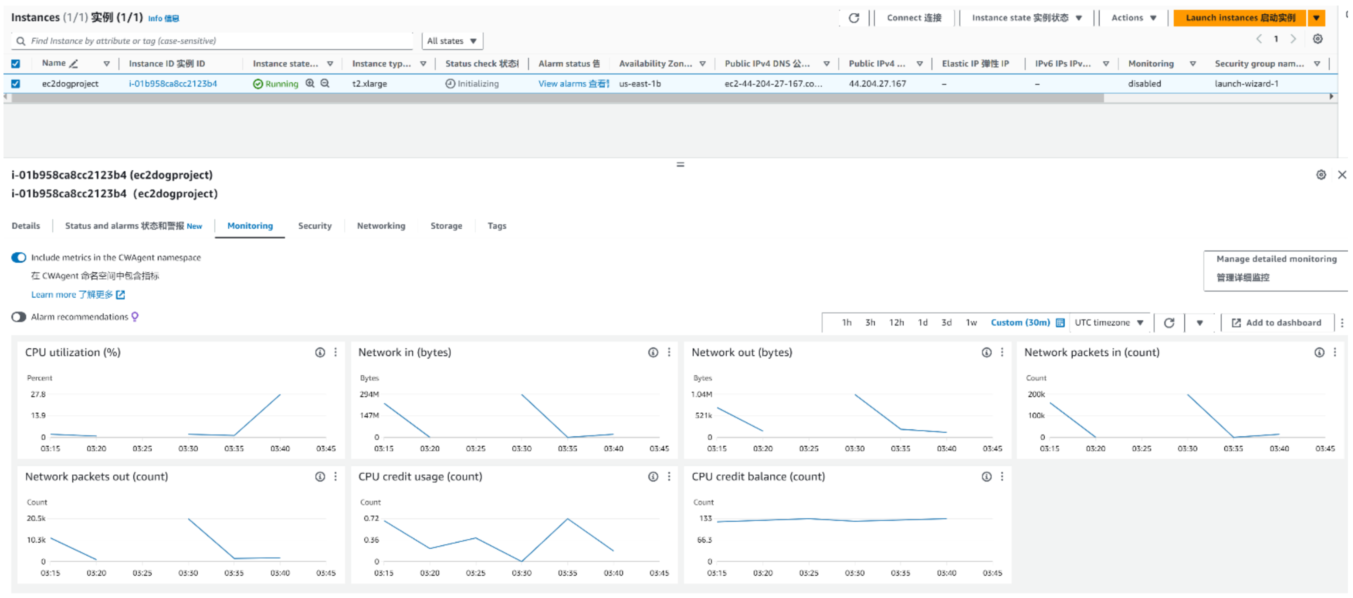


```
[root@ip-172-31-84-235 ~]# source activate pytorch
WARNING: Please note that the Amazon EC2 t2.xlarge instance type is not supported by current Deep Learning AMI.
Please refer the DLAMI release notes https://docs.aws.amazon.com/dlami/latest/devguide/appendix-ami-release-notes.html for more information.
(pytorch) [root@ip-172-31-84-235 ~]# python solution.py
/opt/conda/envs/pytorch/lib/python3.10/site-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please
use 'weights' instead.
  warnings.warn(
/opt/conda/envs/pytorch/lib/python3.10/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may
be removed in the future. The current behavior is equivalent to passing `weights=ResNet50_Weights.IMAGENET1K_V1`. You can also use `weights=ResNet50_Weights.DEFAULT` to get the most up-to-date
weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to /root/.cache/torch/hub/checkpoints/resnet50-0676ba61.pth
100%|████████████████████████████████████████████████████████████████████████| 97.8M/97.8M [00:00<00:00, 137MB/s]
Starting Model Training
saved
(pytorch) [root@ip-172-31-84-235 ~]# ls
dogImages  dogImages.zip  dogImages.zip.1  python.py  solution.py  TrainedModels
(pytorch) [root@ip-172-31-84-235 ~]#
```

i-01b958ca8cc2123b4 (ec2dogproject)

PublicIPs: 44.204.27.167 PrivateIPs: 172.31.84.235

Figure 10. EC2 Training saved - model.pth



Instances (1/1) 实例 (1/1) Info 信息

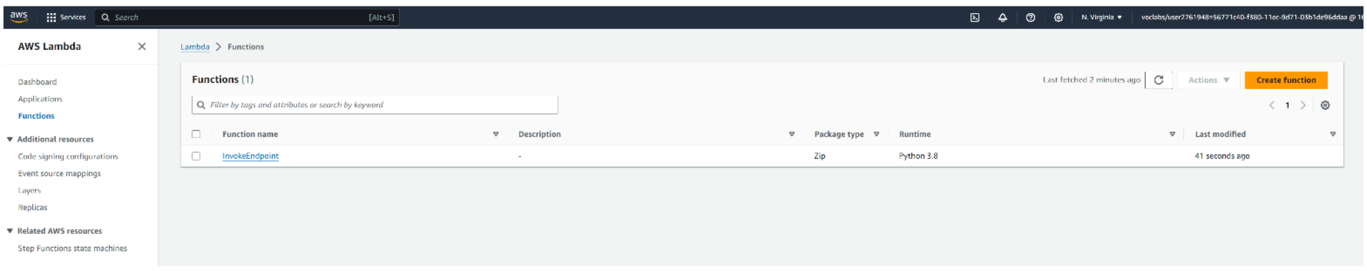
	Name	Instance ID 实例 ID	Instance state...	Instance typ...	Status check 状态	Alarm status 告	Availability Zon...	Public IPv4 DNS 公...	Public IPv4 ...	Elastic IP 弹性 IP	IPv6 IPs IPv...	Monitoring	Security group nam...
✓	ec2dogproject	i-01b958ca8cc2123b4	Running	t2.xlarge	Initializing	View alarms 查看	us-east-1b	ec2-44-204-27-167.co...	44.204.27.167	–	–	disabled	launch-wizard-1

i-01b958ca8cc2123b4 (ec2dogproject)
i-01b958ca8cc2123b4 (ec2dogproject)

Figure 11. EC2 Instance Resource Metrics

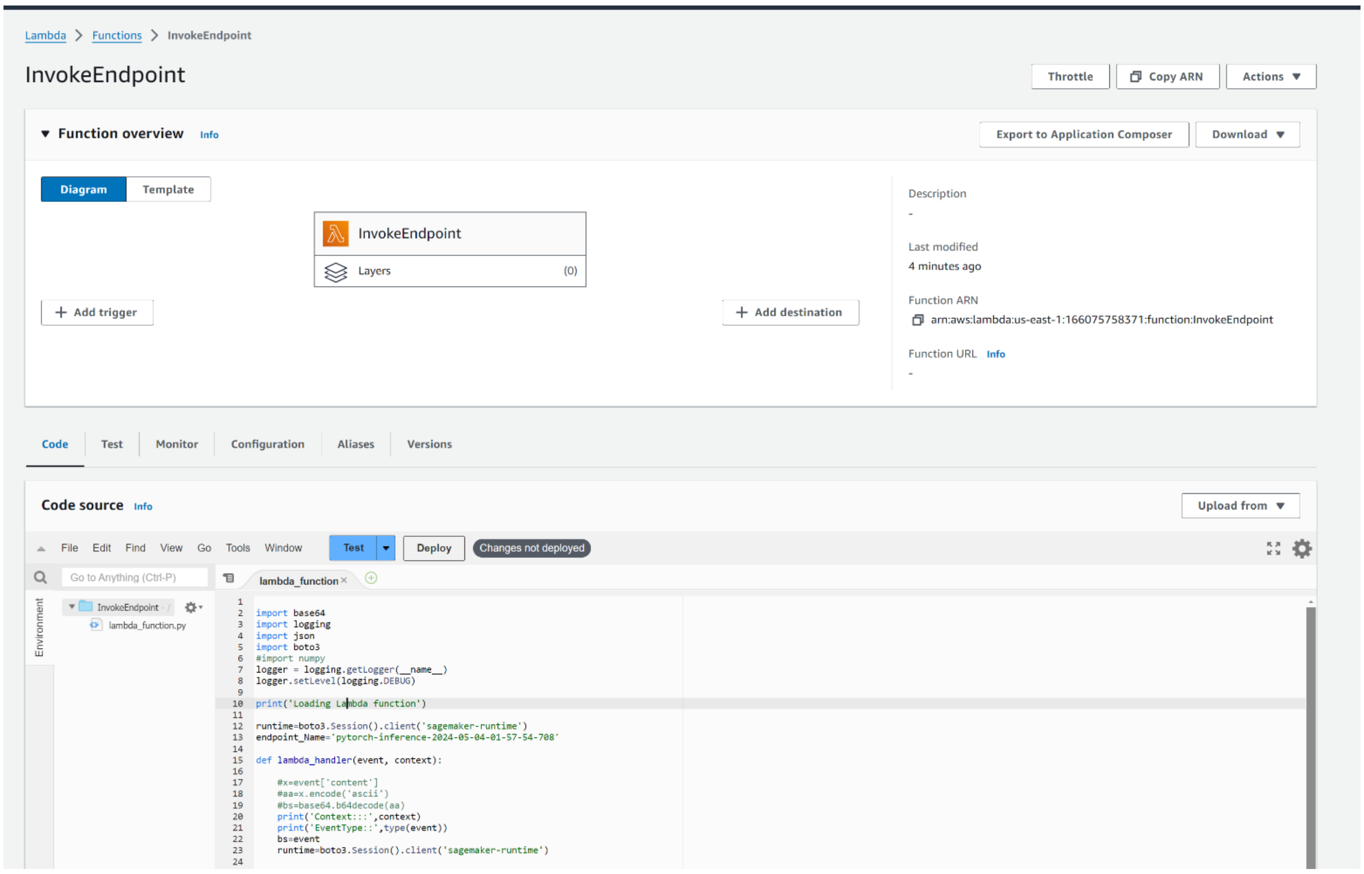
4. Lambda functions

- The lambda functions will be used for invoking our deployed endpoints.
- The lambda function implemented in this project expects the image inputs in json format, which is used to invoke the model's deployed endpoint
- Given we have two endpoints deployed, one for the single instance training and the other for the multi-instance training, we will only use the multi-instance training jobs endpoint and create a lambda function for invoking that endpoint.
- Multi instance trained endpoint that we will be using: "**pytorch-inference-2022-06-08-20-39-47-131**"
- We created the lambda function with the corresponding changes to invoke the endpoint:



Function name	Description	Package type	Runtime	Last modified
InvokeEndpoint	-	Zip	Python 3.8	41 seconds ago

Figure 12. Lambda Function



Description
-

Last modified
4 minutes ago

Function ARN
arn:aws:lambda:us-east-1:166075758371:function:InvokeEndpoint

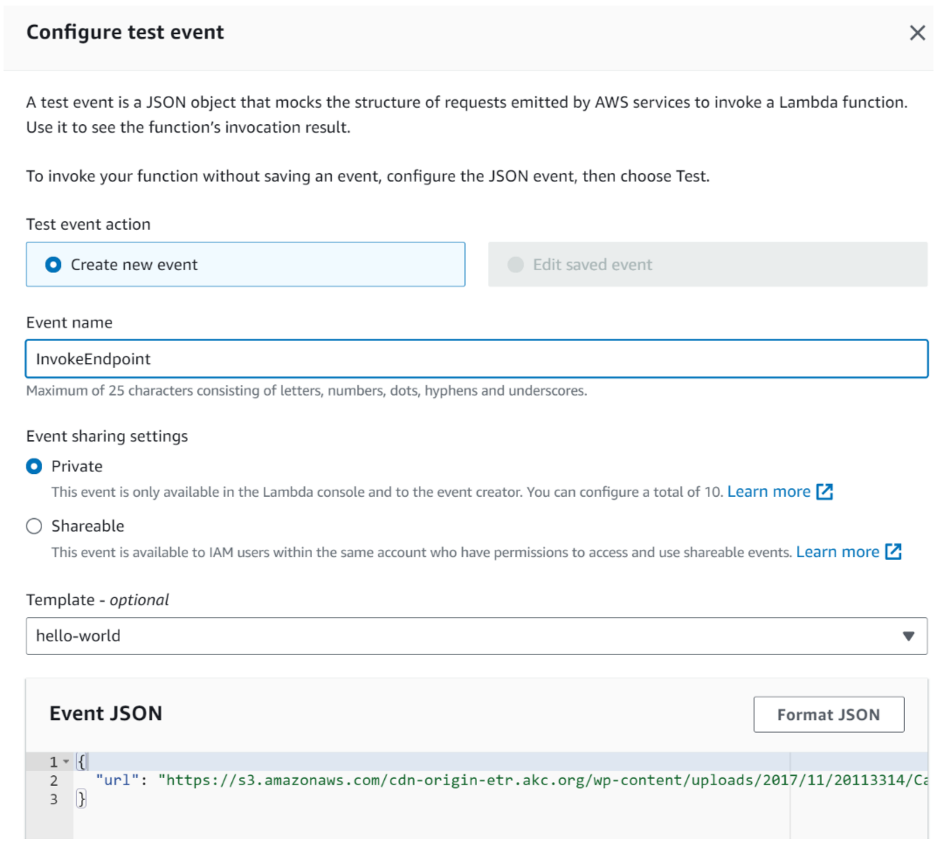
Function URL Info
-

```python

import base64
import logging
import json
import boto3
#import numpy
logger = logging.getLogger(__name__)
logger.setLevel(logging.DEBUG)

print('Loading Lambda function')

runtime=boto3.Session().client('sagemaker-runtime')
endpoint_Name='pytorch-inference-2024-05-04-01-57-54-708'

def lambda_handler(event, context):

    #x=event['content']
    #aa=x.encode('ascii')
    #bs=base64.b64decode(aa)
    print('Context:::',context)
    print('EventType::',type(event))
    bs=event
    runtime=boto3.Session().client('sagemaker-runtime')

```

Figure 13. Lambda Function

5. Security and Testing

- We had created a new role for this lambda functions with basic accesses.
- We used the below test event to test our lambda function:


Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

[✓ Create new event] Edit saved event

Event name

InvokeEndpoint

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

[✓ Private]

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world

Event JSON

Format JSON

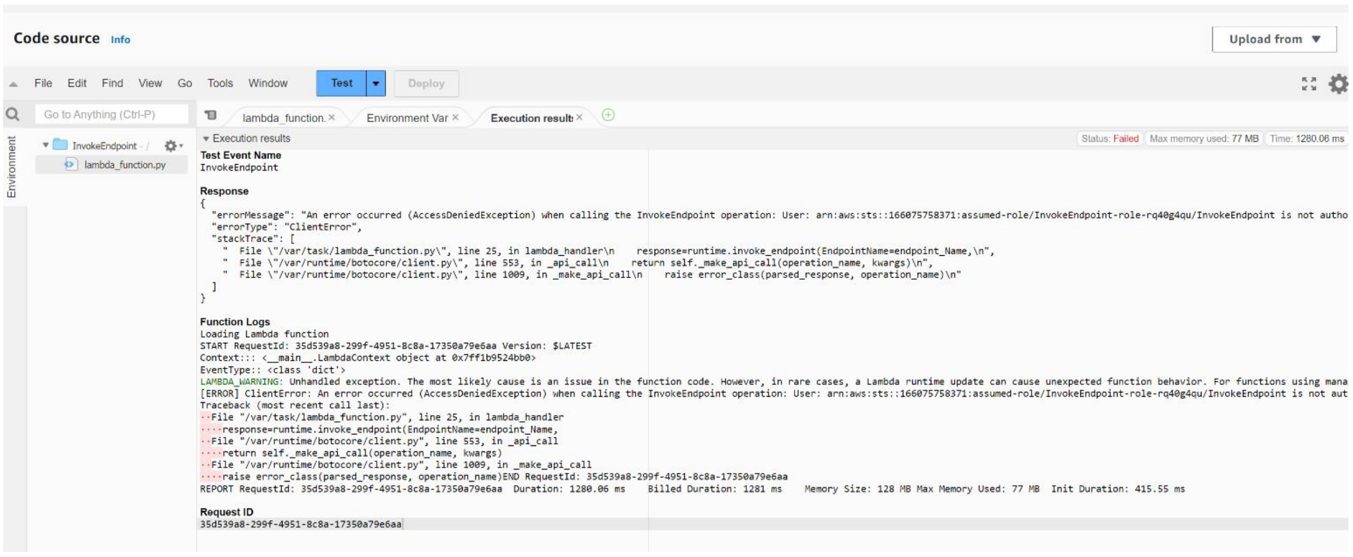
```
{
  "url": "https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2017/11/20113314/Ca
}
```

Figure 14. Lambda Function Test Event

- Now when we tried to execute the test event we got and **AccessDeniedException.** This was expected as the lambda function did not have access to invoke the SageMaker endpoint.

- Error Message:

```
 {
  "errorMessage": "An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::594529116801:assumed-role/invokeEndpoint-role-4qryi0br/invokeEndpoint is not authorized to perform: sagemaker:InvokeEndpoint on resource: arn:aws:sagemaker:us-east-1:594529116801:endpoint/pytorch-inference-2024-05-03-10-39-47-131 because no identity-based policy allows the sagemaker:InvokeEndpoint action",
  "errorType": "ClientError",
  "stackTrace": [
    "  File \"/var/task/lambda_function.py\", line 24, in lambda_handler\n    response=runtime.invoke_endpoint(EndpointName=endpoint_Name,\n",
    "  File \"/var/runtime/botocore/client.py\", line 391, in _api_call\n    return self._make_api_call(operation_name, kwargs)\n",
    "  File \"/var/runtime/botocore/client.py\", line 719, in _make_api_call\n    raise error_class(parsed_response, operation_name)\n"
  ]
}
```



Status: Failed | Max memory used: 77 MB | Time: 1280.06 ms

Test Event Name
InvokeEndpoint

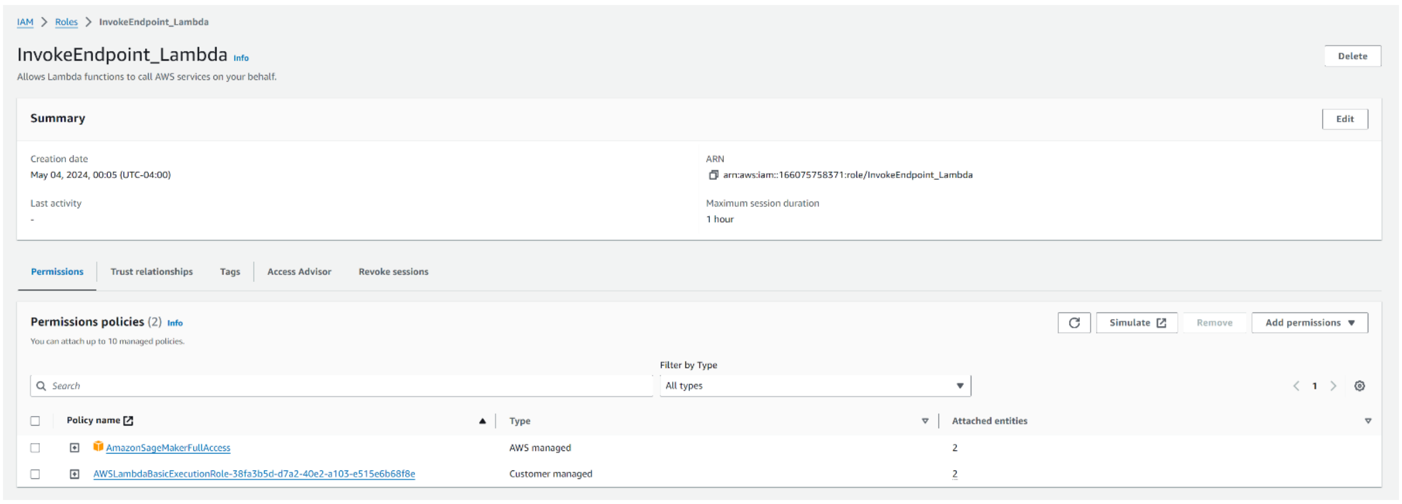
Response
```
{
  "errorMessage": "An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::166075758371:assumed-role/InvokeEndpoint-role-rq40g4qu/InvokeEndpoint is not autho
  "errorType": "ClientError",
  "stackTrace": [
    "  File \"/var/task/lambda_function.py\", line 25, in lambda_handler\n    response=runtime.invoke_endpoint(EndpointName=endpoint_Name,\n",
    "  File \"/var/runtime/botocore/client.py\", line 553, in _api_call\n    return self._make_api_call(operation_name, kwargs)\n",
    "  File \"/var/runtime/botocore/client.py\", line 1009, in _make_api_call\n    raise error_class(parsed_response, operation_name)\n"
  ]
}
```

Function Logs
```
Loading Lambda function
START RequestId: 35d539a8-299f-4951-8c8a-17350a79e6aa Version: $LATEST
Context::: <__main__.LambdaContext object at 0x7ff1b9524bb0>
EventType:: <class 'dict'>
LAMBDA_WARNING: Unhandled exception. The most likely cause is an issue in the function code. However, in rare cases, a Lambda runtime update can cause unexpected function behavior. For functions using mana
[ERROR] ClientError: An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::166075758371:assumed-role/InvokeEndpoint-role-rq40g4qu/InvokeEndpoint is not aut
Traceback (most recent call last):
  File "/var/task/lambda_function.py", line 25, in lambda_handler
    response=runtime.invoke_endpoint(EndpointName=endpoint_Name,
  File "/var/runtime/botocore/client.py", line 553, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "/var/runtime/botocore/client.py", line 1009, in _make_api_call
    raise error_class(parsed_response, operation_name)END RequestId: 35d539a8-299f-4951-8c8a-17350a79e6aa
REPORT RequestId: 35d539a8-299f-4951-8c8a-17350a79e6aa  Duration: 1280.06 ms    Billed Duration: 1281 ms    Memory Size: 128 MB Max Memory Used: 77 MB  Init Duration: 415.55 ms
```

Request ID
35d539a8-299f-4951-8c8a-17350a79e6aa

Figure 15. Lambda function test event failure response

- So went ahead and added the “**SageMakerFullAccess”** policy role to the lambda function’s role.



InvokeEndpoint_Lambda

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
May 04, 2024, 00:05 (UTC-04:00)

Last activity
-

ARN
arn:aws:iam::166075758371:role/InvokeEndpoint_Lambda

Maximum session duration
1 hour

Permissions policies (2)

You can attach up to 10 managed policies.

Filter by Type: All types

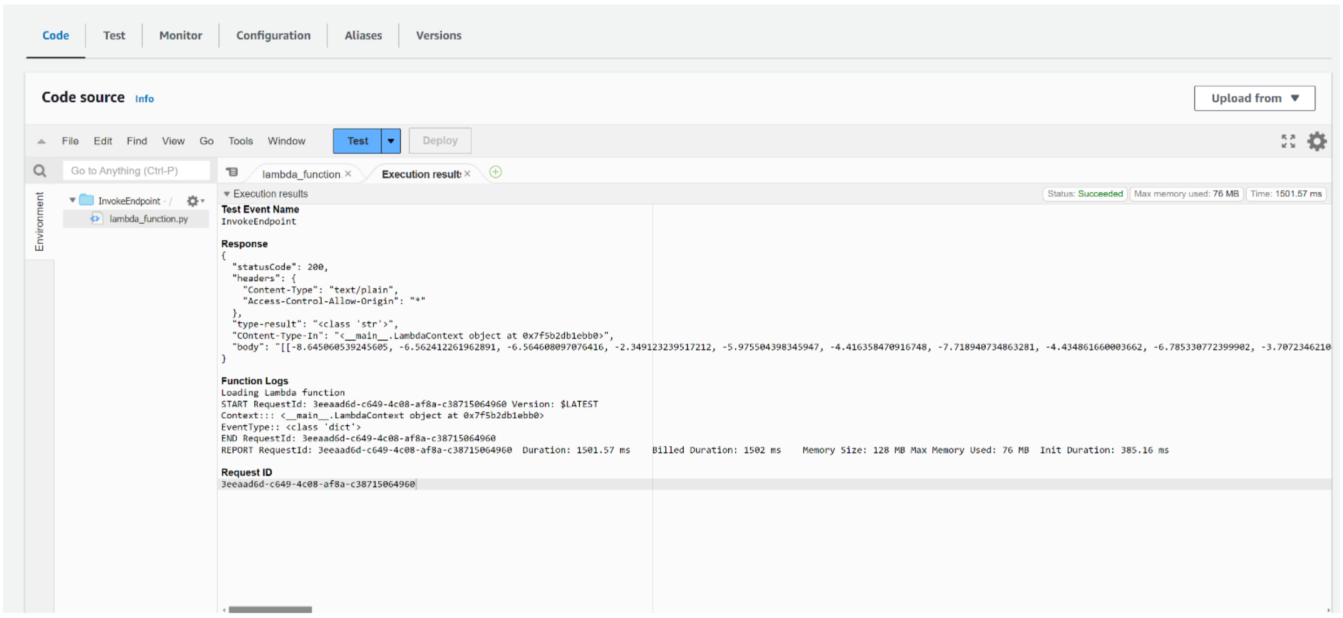
Policy name	Type	Attached entities
AmazonSageMakerFullAccess	AWS managed	2
AWSLambdaBasicExecutionRole-38fa3b5d-d7a2-40e2-a103-e515e6b68f8e	Customer managed	2

Figure 16. Lambda function role IAM permissions

- Post which we got the following output from the test event:
- (Please note that there are 133 dog breed and not 33 as mentioned in the project instructions. So, we do expect there to be around 133 elements in the response object.)
- Response:

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "<__main__.LambdaContext object at 0x7f3802e37b20>",
  "body": "[[-7.732433795928955, -7.176344394683838, -6.378891468048096, -1.5176353454589844, -
3.4210686683654785, -9.79492473602295, -6.717096328735352, -1.7850936651229858, -
8.523921012878418, -1.404198169708252, -0.7228893637657166, -5.840402603149414, -
5.7331461906433105, -2.0368082523345947, -9.905051231384277, -6.480345726013184, -
10.379494667053223, -2.0244674682617188, -10.030835151672363, 0.24102921783924103, -
7.826473712921143, -2.3736279010772705, -7.411016464233398, -9.585257530212402, -
```

6.935138702392578, -10.370619773864746, -6.9618120193481445, -5.8000054359436035, -7.852663516998291, -4.261888027191162, -8.121268272399902, -2.3057727813720703, -9.783154487609863, -4.807986736297607, -8.92819595336914, -8.51605224609375, -8.291374206542969, -5.918830394744873, -3.329312324523926, -7.71879768371582, -4.240849018096924, -8.23236083984375, -0.6575157642364502, -3.2674717903137207, -1.3542909622192383, -12.321500778198242, -5.892731666564941, -0.7067917585372925, -4.4090352058410645, -3.0732972621917725, -3.965545892715454, -12.868936538696289, -10.151001930236816, -5.127017974853516, -9.019100189208984, -2.250539779663086, -6.182336807250977, -10.275800704956055, -1.599454641342163, -6.101325988769531, -13.15511417388916, -13.388274192810059, -12.73229694366455, -6.6955180168151855, -3.183396577835083, -13.372259140014648, -2.6620848178863525, -5.736507892608643, -3.5820565223693848, -0.9081988334655762, 1.058609127998352, -7.375503063201904, -7.269904136657715, -5.875174045562744, -8.469480514526367, -4.669737815856934, -8.637510299682617, -2.2320778369903564, -6.134050369262695, -4.5598530769348145, -2.207737922668457, -10.790899276733398, -2.3271493911743164, -3.509336471557617, -10.742084503173828, -11.33560848236084, -7.614548683166504, -12.094978332519531, -5.790018558502197, -2.1963579654693604, -9.642144203186035, -7.176695346832275, -6.40428352355957, -11.0121488571167, -6.617325305938721, -3.0795605182647705, -7.987880706787109, -3.7228918075561523, -9.426275253295898, -7.498056888580322, -11.697783470153809, -1.4410004615783691, -2.5681569576263428, -4.899290084838867, -3.1299190521240234, -5.716061115264893, -8.185333251953125, -3.402395248413086, -2.385983467102051, -2.442678928375244, -8.68392562866211, -2.8638479709625244, -11.222406387329102, -10.302404403686523, -7.736847877502441, -4.212231636047363, -7.31355619430542, -4.08905029296875, -12.546906471252441, -2.2557764053344727, -2.3533856868743896, -6.100186347961426, -7.259090900421143, -6.4662346839904785, -7.800014495849609, -6.287154197692871, -6.792305946350098, -2.09424090385437, -4.197015762329102, -12.041462898254395, -10.169670104980469, -1.5505112409591675, -6.9473676681518555]]"
}

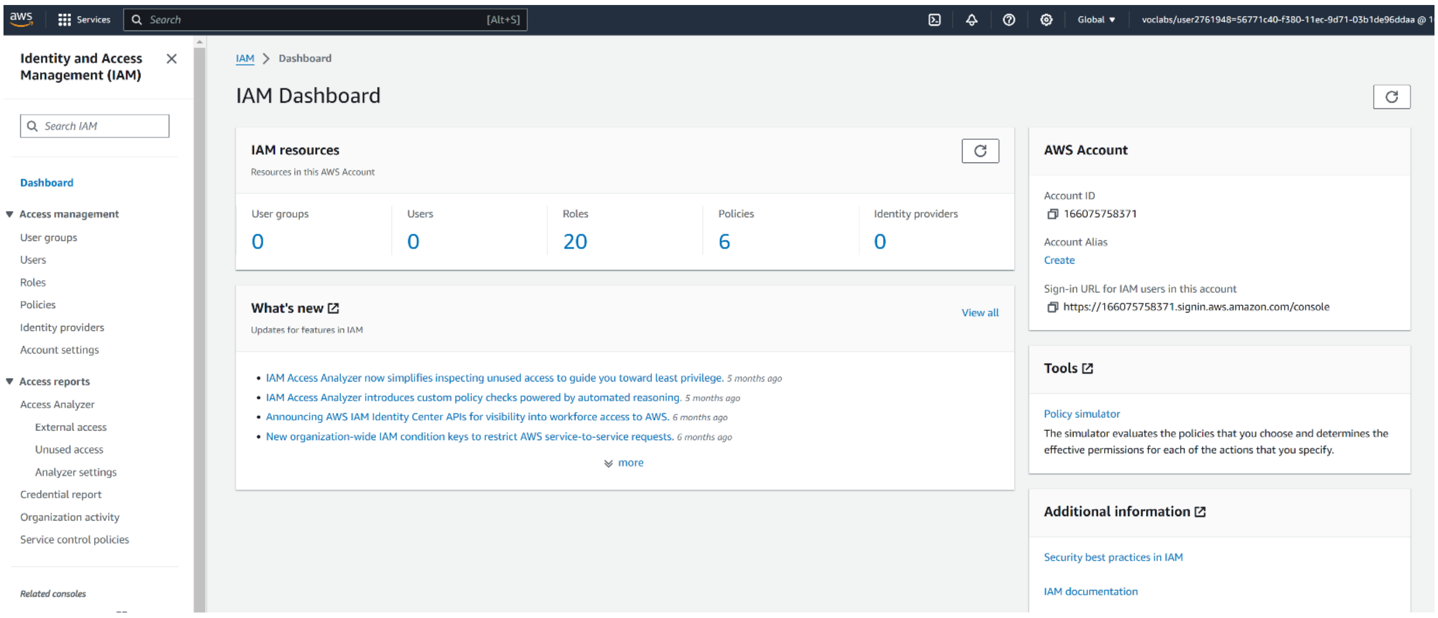


```
Execution results
Test Event Name
InvokeEndpoint

Response
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "<__main__.LambdaContext object at 0x7f5b2db1ebb0>",
  "body": "[[-8.645060539245605, -6.562412261962891, -6.564608097076416, -2.349123239517212, -5.975504398345947, -4.416358470916748, -7.718940734863281, -4.434861660003662, -6.785330772399902, -3.7072346210
}

Function Logs
Loading Lambda function
START RequestId: 3eeaad6d-c649-4c08-af8a-c38715064960 Version: $LATEST
Context::: <__main__.LambdaContext object at 0x7f5b2db1ebb0>
EventType:: <class 'dict'>
END RequestId: 3eeaad6d-c649-4c08-af8a-c38715064960
REPORT RequestId: 3eeaad6d-c649-4c08-af8a-c38715064960  Duration: 1501.57 ms    Billed Duration: 1502 ms    Memory Size: 128 MB Max Memory Used: 76 MB  Init Duration: 385.16 ms

Request ID
3eeaad6d-c649-4c08-af8a-c38715064960
```

Figure 17. Lambda function test event success response

- I'm concerned about the “Full Access” type permission policies that are available.
- For example, for this lambda functions we have provided the lambda function a Full Access to SageMaker resources, but this does not seem to follow the concept of least privilege access.
- Ideally, one should only allow these lambda functions to query the endpoints that they're intended and allowed to query.

- We will have to do some more analysis to figure out if there's anything we could do about it.
- Furthermore, another concern is that the account's root user does not employ MFA
- Looking at the IAM roles that are currently active, all the roles seem to be necessary and most of the roles have been added on a per need basis.
- However, we need to keep an eye on the roles dashboard to make sure only relevant roles necessary for currently active projects, are the only roles that are in active state to prevent unauthorized accesses.



IAM Dashboard

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	20	6	0

What's new

Updates for features in IAM

View all

- IAM Access Analyzer now simplifies inspecting unused access to guide you toward least privilege. *5 months ago*
- IAM Access Analyzer introduces custom policy checks powered by automated reasoning. *5 months ago*
- Announcing AWS IAM Identity Center APIs for visibility into workforce access to AWS. *6 months ago*
- New organization-wide IAM condition keys to restrict AWS service-to-service requests. *6 months ago*

more

AWS Account

Account ID
166075758371

Account Alias
Create

Sign-in URL for IAM users in this account
https://166075758371.signin.aws.amazon.com/console

Tools

Policy simulator

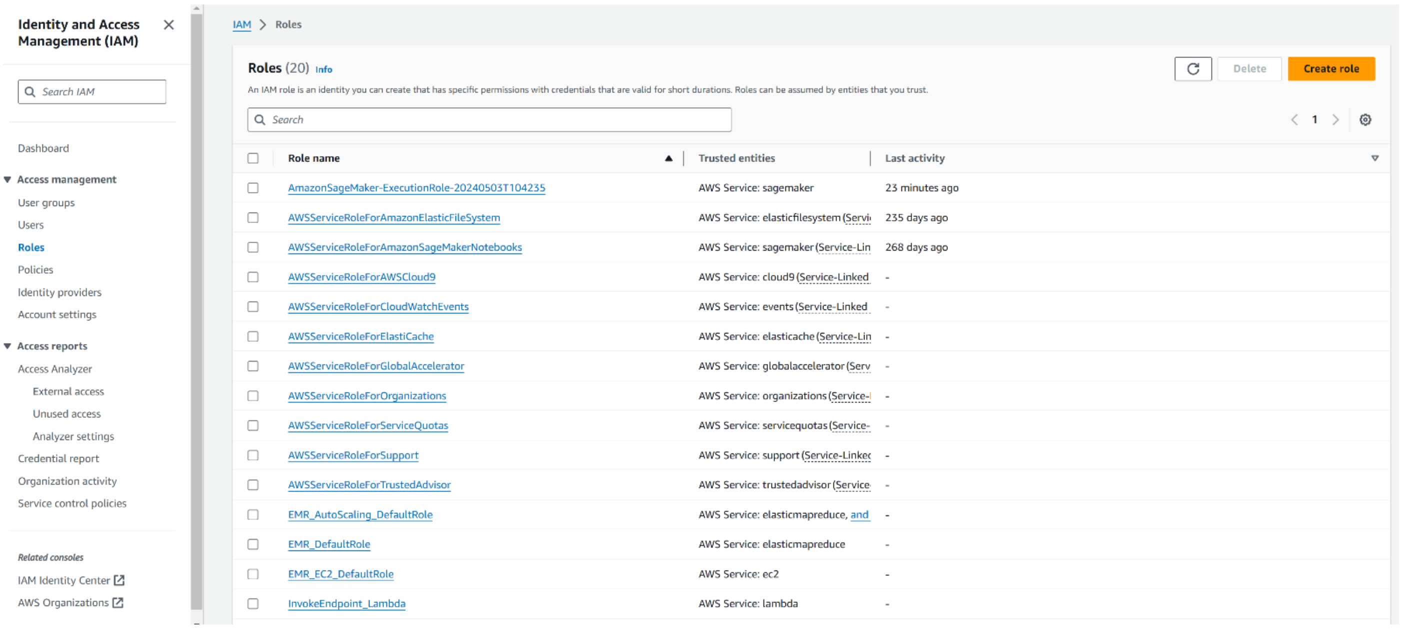
The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.

Additional information

Security best practices in IAM

IAM documentation

Figure 18. IAM Dashboard



Roles (20) Info

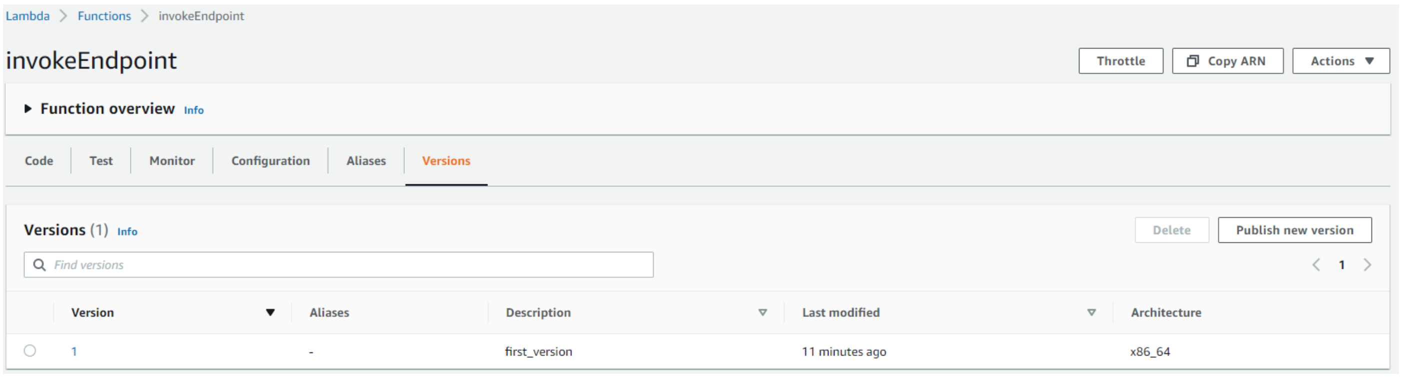
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AmazonSageMaker-ExecutionRole-20240503T104235	AWS Service: sagemaker	23 minutes ago
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Servi	235 days ago
AWSServiceRoleForAmazonSageMakerNotebooks	AWS Service: sagemaker (Service-Lin	268 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked	-
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked	-
AWSServiceRoleForElastiCache	AWS Service: elasticache (Service-Lin	-
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Serv	-
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-l	-
AWSServiceRoleForServiceQuotas	AWS Service: servicequotas (Service-	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
EMR_AutoScaling_DefaultRole	AWS Service: elasticmapreduce, and	-
EMR_DefaultRole	AWS Service: elasticmapreduce	-
EMR_EC2_DefaultRole	AWS Service: ec2	-
InvokeEndpoint_Lambda	AWS Service: lambda	-

Figure 19. IAM Roles

6. Concurrency and Auto-scaling

- Before adding in configs for Concurrency and Auto-scaling for our lambda functions we will first create a version config for our lambda function.


Lambda > Functions > invokeEndpoint

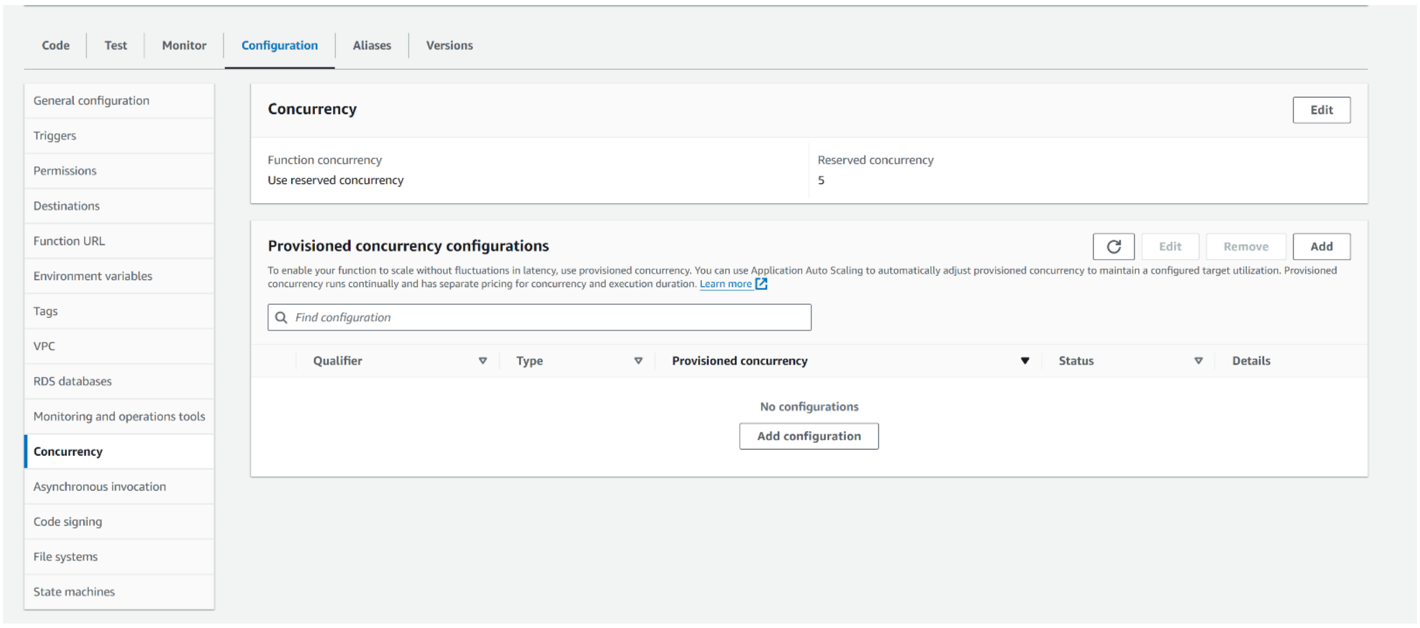
invokeEndpoint

Function overview

Version	Aliases	Description	Last modified	Architecture
1	-	first_version	11 minutes ago	x86_64

Figure 20. Lambda function version config

- For the lambda function we have set the **reserved concurrency** to be 5. This implies that the lambda function would be able to handle up to 5 requests concurrently at the same time. This would help lower latency issues in situations when there is higher traffic than usual.


Concurrency

Function concurrency: Use reserved concurrency

Reserved concurrency: 5

Provisioned concurrency configurations

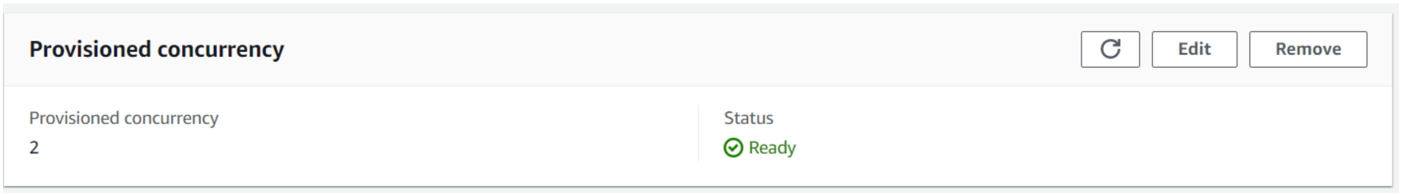
To enable your function to scale without fluctuations in latency, use provisioned concurrency. You can use Application Auto Scaling to automatically adjust provisioned concurrency to maintain a configured target utilization. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. Learn more

Qualifier	Type	Provisioned concurrency	Status	Details

No configurations

Figure 21. lambda function reversed concurrency

- Given the current use case, ideally using only reserved concurrency should have sufficed for our use case and we might not need to consider using the provisioned concurrency configs. However, for the sake of completion, I tried to add in the config for the provisioned concurrency as well. We set the provision concurrency to be 2.

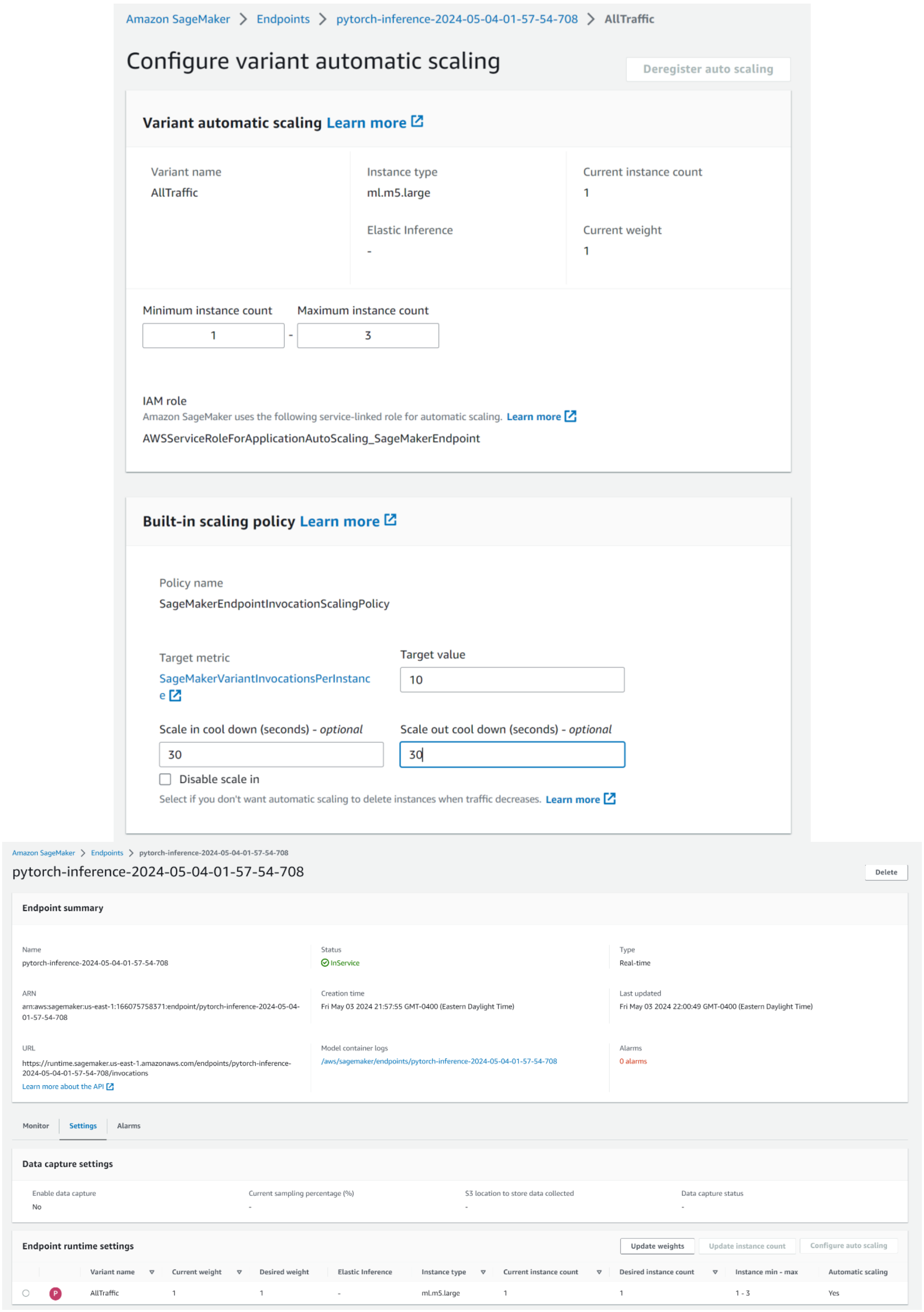

Provisioned concurrency

Provisioned concurrency: 2

Status: Ready

Figure 22. lambda function provisioned concurrency

- For adding config for auto-scaling we have added the below configs:



Amazon SageMaker > Endpoints > pytorch-inference-2024-05-04-01-57-54-708 > AllTraffic

Configure variant automatic scaling

Deregister auto scaling

Variant automatic scaling Learn more

Variant name	Instance type	Current instance count
AllTraffic	ml.m5.large	1
	Elastic Inference	Current weight
	-	1

Minimum instance count: 1 - Maximum instance count: 3

IAM role

Amazon SageMaker uses the following service-linked role for automatic scaling. Learn more

AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint

Built-in scaling policy Learn more

Policy name: SageMakerEndpointInvocationScalingPolicy

Target metric: SageMakerVariantInvocationsPerInstance

Target value: 10

Scale in cool down (seconds) - *optional*: 30

Scale out cool down (seconds) - *optional*: 30

☐ Disable scale in

Select if you don't want automatic scaling to delete instances when traffic decreases. Learn more

Amazon SageMaker > Endpoints > pytorch-inference-2024-05-04-01-57-54-708

pytorch-inference-2024-05-04-01-57-54-708

Delete

Endpoint summary

Name	Status	Type
pytorch-inference-2024-05-04-01-57-54-708	InService	Real-time
ARN	Creation time	Last updated
arn:aws:sagemaker:us-east-1:166075758371:endpoint/pytorch-inference-2024-05-04-01-57-54-708	Fri May 03 2024 21:57:55 GMT-0400 (Eastern Daylight Time)	Fri May 03 2024 22:00:49 GMT-0400 (Eastern Daylight Time)
URL	Model container logs	Alarms
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2024-05-04-01-57-54-708/invocations Learn more about the API	/aws/sagemaker/endpoints/pytorch-inference-2024-05-04-01-57-54-708	0 alarms

Monitor | Settings | Alarms

Data capture settings

Enable data capture	Current sampling percentage (%)	S3 location to store data collected	Data capture status
No	-	-	-

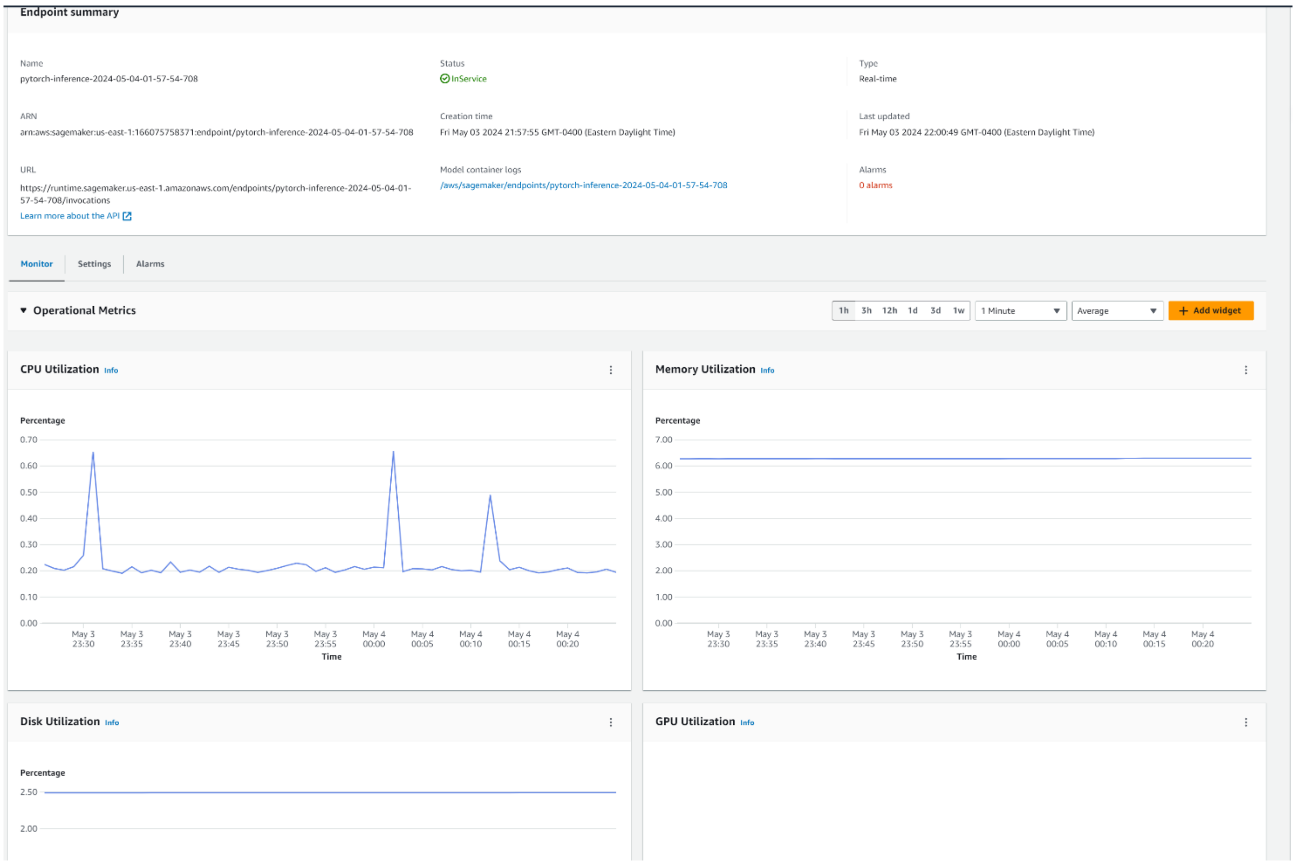
Endpoint runtime settings

Update weights | Update instance count | Configure auto scaling

		Variant name	Current weight	Desired weight	Elastic Inference	Instance type	Current instance count	Desired instance count	Instance min - max	Automatic scaling
○	P	AllTraffic	1	1	-	ml.m5.large	1	1	1 - 3	Yes

Figure 23. Endpoint Auto-scaling Config

- We have set the max instance count to 3 for Auto-scaling, as considering the current requirement, auto scaling on 3 instances with a scale-in and scale-out cool down time of 30 seconds should be a reasonably good config.



Endpoint summary

Name
pytorch-inference-2024-05-04-01-57-54-708

Status
InService

Type
Real-time

ARN
arn:aws:sagemaker:us-east-1:166075758371:endpoint/pytorch-inference-2024-05-04-01-57-54-708

Creation time
Fri May 03 2024 21:57:55 GMT-0400 (Eastern Daylight Time)

Last updated
Fri May 03 2024 22:00:49 GMT-0400 (Eastern Daylight Time)

URL
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2024-05-04-01-57-54-708/invocations

Model container logs
/aws/sagemaker/endpoints/pytorch-inference-2024-05-04-01-57-54-708

Alarms
0 alarms

Figure 24. Endpoint Auto-scaling Metrics