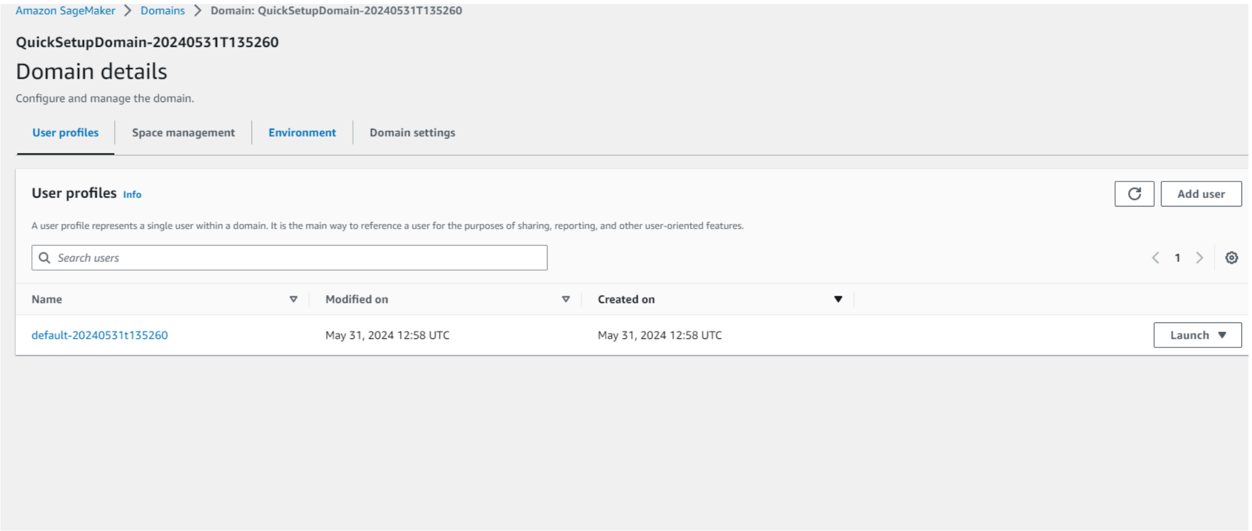
Project: Operationalizing an AWS ML Project

Step 1: Training and deployment on Sagemaker

I choosed the default quick setup domain which has all the jupyter notebooks and configurations set up , i have used simple fast launch instances as they are cost effective, i have updated the cells with my s3 bucket name and also created the multi-instance training estimator and have also checked for the best hyper parameters

AWS sagemaker setup



QuickSetupDomain-20240531T135260

Domain details

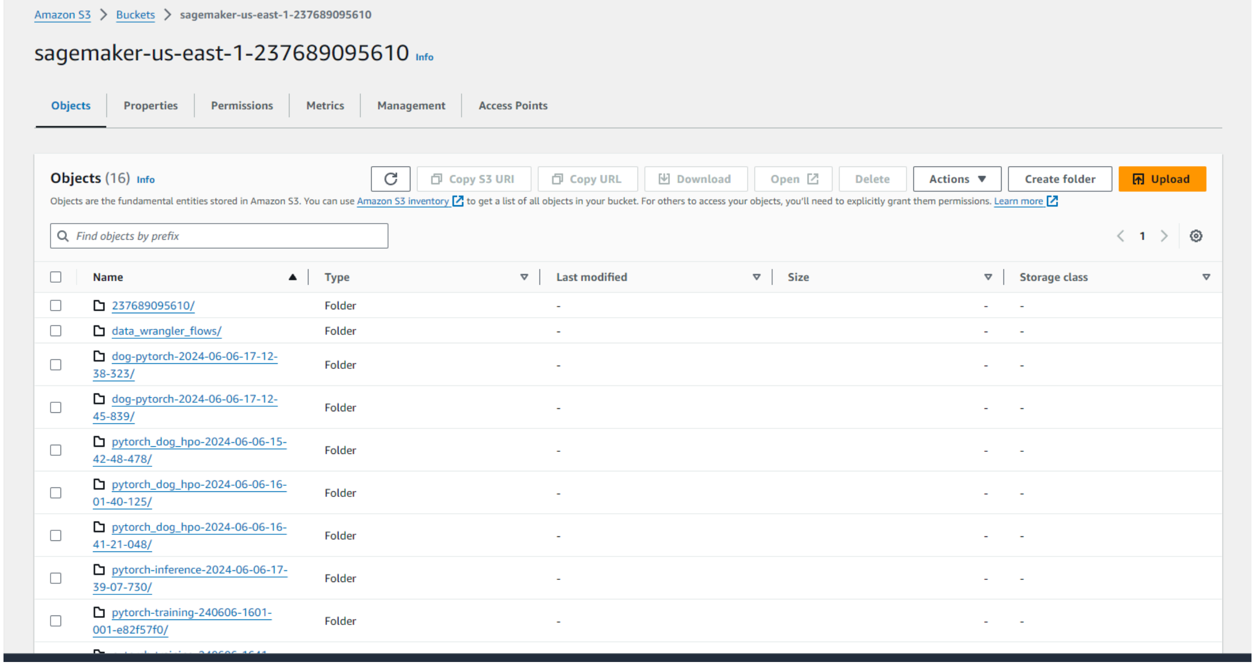
Configure and manage the domain.

User profiles

A user profile represents a single user within a domain. It is the main way to reference a user for the purposes of sharing, reporting, and other user-oriented features.

Name	Modified on	Created on
default-20240531t135260	May 31, 2024 12:58 UTC	May 31, 2024 12:58 UTC

S3 Bucket



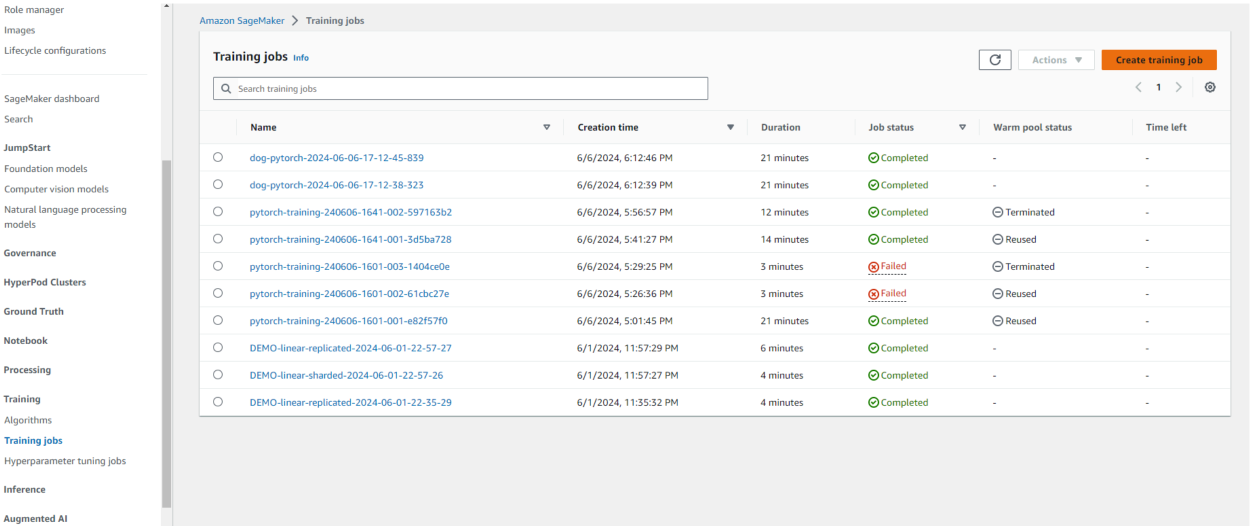
sagemaker-us-east-1-237689095610

Objects (16)

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
☐	237689095610/	Folder	-	-	-
☐	data_wrangler_flows/	Folder	-	-	-
☐	dog-pytorch-2024-06-06-17-12-38-323/	Folder	-	-	-
☐	dog-pytorch-2024-06-06-17-12-45-839/	Folder	-	-	-
☐	pytorch_dog_hpo-2024-06-06-15-42-48-478/	Folder	-	-	-
☐	pytorch_dog_hpo-2024-06-06-16-01-40-125/	Folder	-	-	-
☐	pytorch_dog_hpo-2024-06-06-16-41-21-048/	Folder	-	-	-
☐	pytorch-inference-2024-06-06-17-39-07-730/	Folder	-	-	-
☐	pytorch-training-240606-1601-001-e82f57f0/	Folder	-	-	-

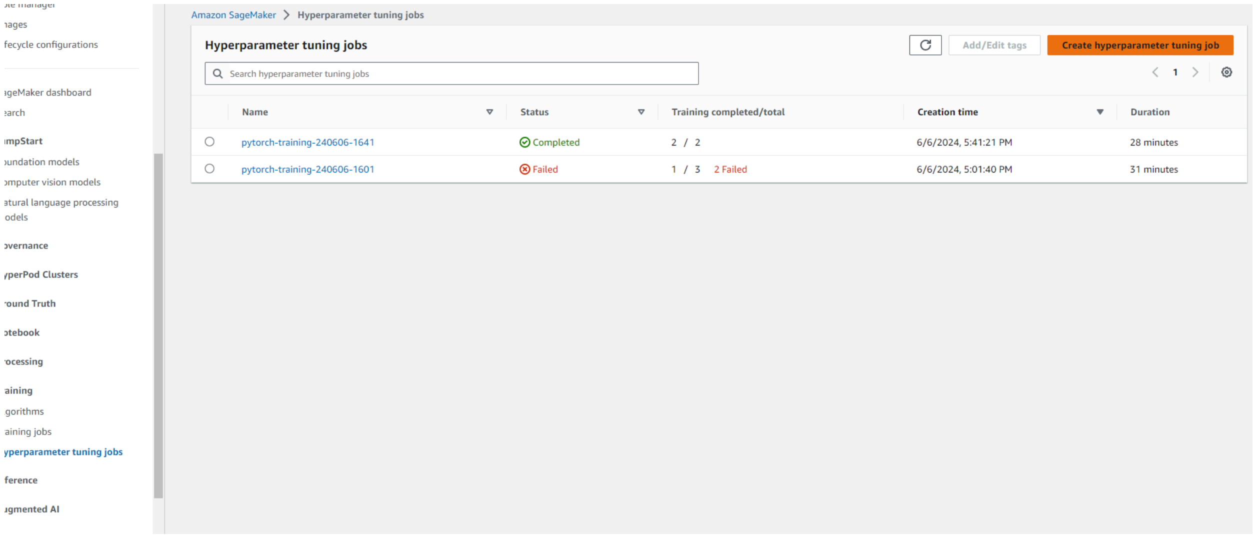
Training Jobs Completed:



Training jobs

	Name	Creation time	Duration	Job status	Warm pool status	Time left
○	dog-pytorch-2024-06-06-17-12-45-839	6/6/2024, 6:12:46 PM	21 minutes	✓ Completed	-	-
○	dog-pytorch-2024-06-06-17-12-38-323	6/6/2024, 6:12:39 PM	21 minutes	✓ Completed	-	-
○	pytorch-training-240606-1641-002-597163b2	6/6/2024, 5:56:57 PM	12 minutes	✓ Completed	⊖ Terminated	-
○	pytorch-training-240606-1641-001-3d5ba728	6/6/2024, 5:41:27 PM	14 minutes	✓ Completed	⊖ Reused	-
○	pytorch-training-240606-1601-003-1404ce0e	6/6/2024, 5:29:25 PM	3 minutes	⊗ Failed	⊖ Terminated	-
○	pytorch-training-240606-1601-002-61cbc27e	6/6/2024, 5:26:36 PM	3 minutes	⊗ Failed	⊖ Reused	-
○	pytorch-training-240606-1601-001-e82f57f0	6/6/2024, 5:01:45 PM	21 minutes	✓ Completed	⊖ Reused	-
○	DEMO-linear-replicated-2024-06-01-22-57-27	6/1/2024, 11:57:29 PM	6 minutes	✓ Completed	-	-
○	DEMO-linear-sharded-2024-06-01-22-57-26	6/1/2024, 11:57:27 PM	4 minutes	✓ Completed	-	-
○	DEMO-linear-replicated-2024-06-01-22-35-29	6/1/2024, 11:35:32 PM	4 minutes	✓ Completed	-	-

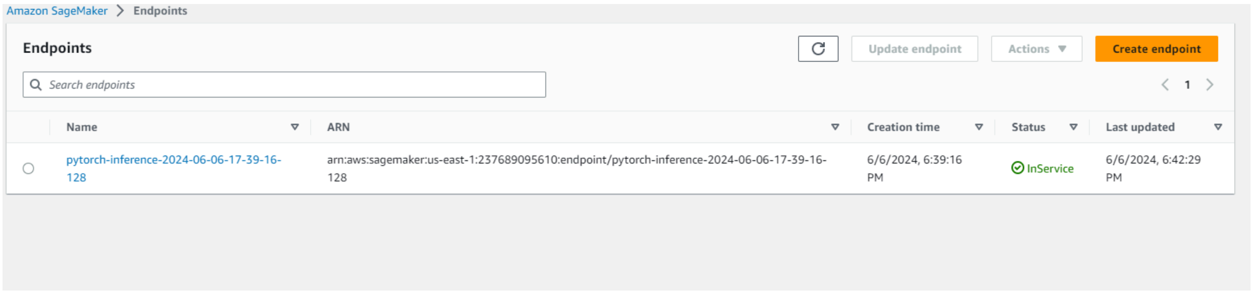
Training Job hyperparamters completed



Hyperparameter tuning jobs

	Name	Status	Training completed/total	Creation time	Duration
○	pytorch-training-240606-1641	Completed	2 / 2	6/6/2024, 5:41:21 PM	28 minutes
○	pytorch-training-240606-1601	Failed	1 / 3 2 Failed	6/6/2024, 5:01:40 PM	31 minutes

Endpoints in service



Endpoints

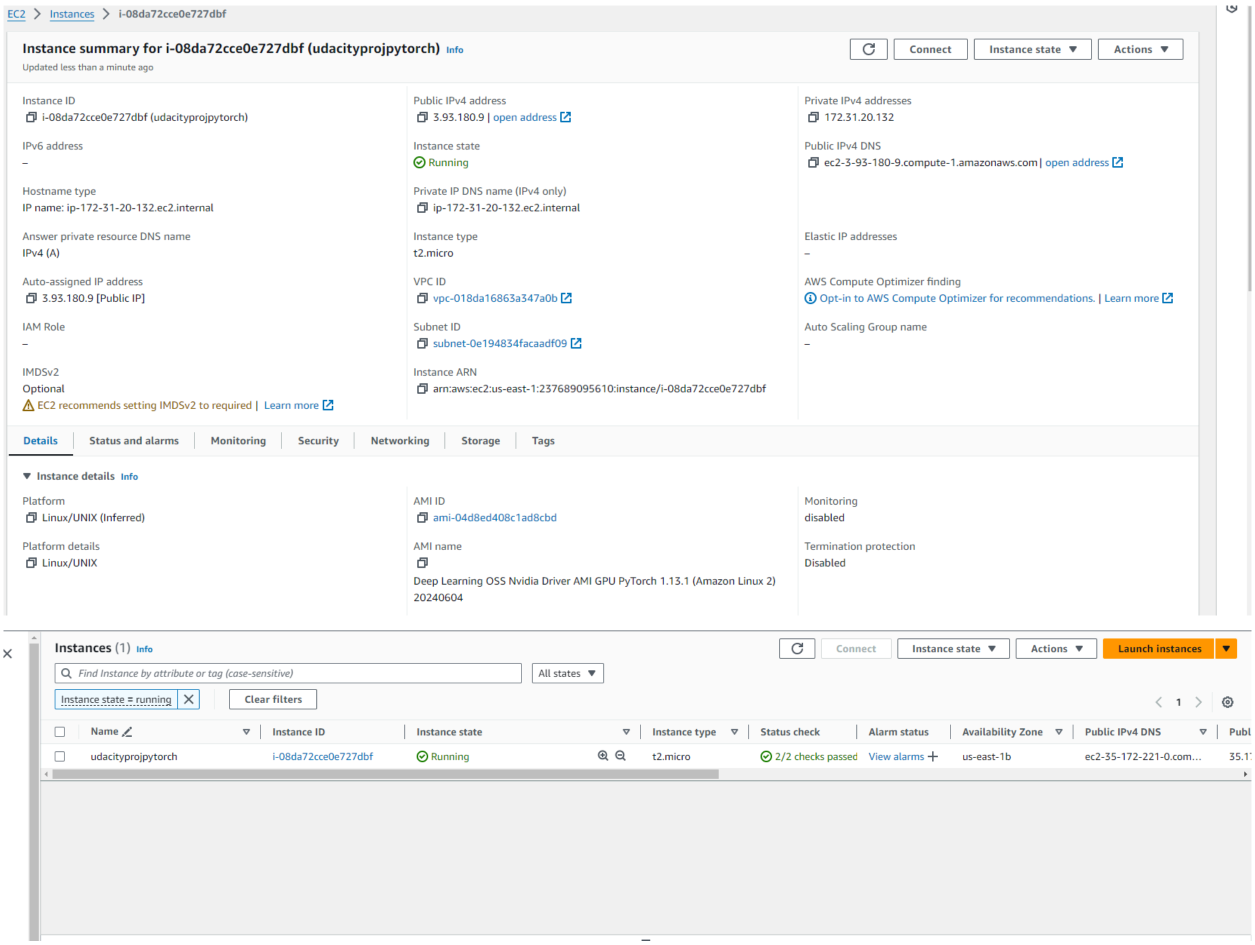
	Name	ARN	Creation time	Status	Last updated
○	pytorch-inference-2024-06-06-17-39-16-128	arn:aws:sagemaker:us-east-1:237689095610:endpoint/pytorch-inference-2024-06-06-17-39-16-128	6/6/2024, 6:39:16 PM	InService	6/6/2024, 6:42:29 PM

Step 2: EC2 Training

Write about ec2 instance created and justification why you choose that instance?

Instance i choose was Deep Learning OSS Nvidia Driver AMI GPU PyTorch 1.13.1 (Amazon Linux 2) 20240604 because this instance has configurations and enough space for running torch library , EBS storage was chosen and that was set to 45GB just so that all libraries could be installed

EC2 Setup



Instance summary for i-08da72cce0e727dbf (udacityprojpytorch) Info

Updated less than a minute ago

Instance ID
i-08da72cce0e727dbf (udacityprojpytorch)

IPv6 address
–

Hostname type
IP name: ip-172-31-20-132.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
3.93.180.9 [Public IP]

IAM Role
–

IMDSv2
Optional
⚠ EC2 recommends setting IMDSv2 to required | Learn more

Public IPv4 address
3.93.180.9 | open address

Instance state
✓ Running

Private IP DNS name (IPv4 only)
ip-172-31-20-132.ec2.internal

Instance type
t2.micro

VPC ID
vpc-018da16863a347a0b

Subnet ID
subnet-0e194834facaadf09

Instance ARN
arn:aws:ec2:us-east-1:237689095610:instance/i-08da72cce0e727dbf

Private IPv4 addresses
172.31.20.132

Public IPv4 DNS
ec2-3-93-180-9.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
ⓘ Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

▼ **Instance details** Info

Platform
Linux/UNIX (Inferred)

Platform details
Linux/UNIX

AMI ID
ami-04d8ed408c1ad8cbd

AMI name
Deep Learning OSS Nvidia Driver AMI GPU PyTorch 1.13.1 (Amazon Linux 2) 20240604

Monitoring
disabled

Termination protection
Disabled

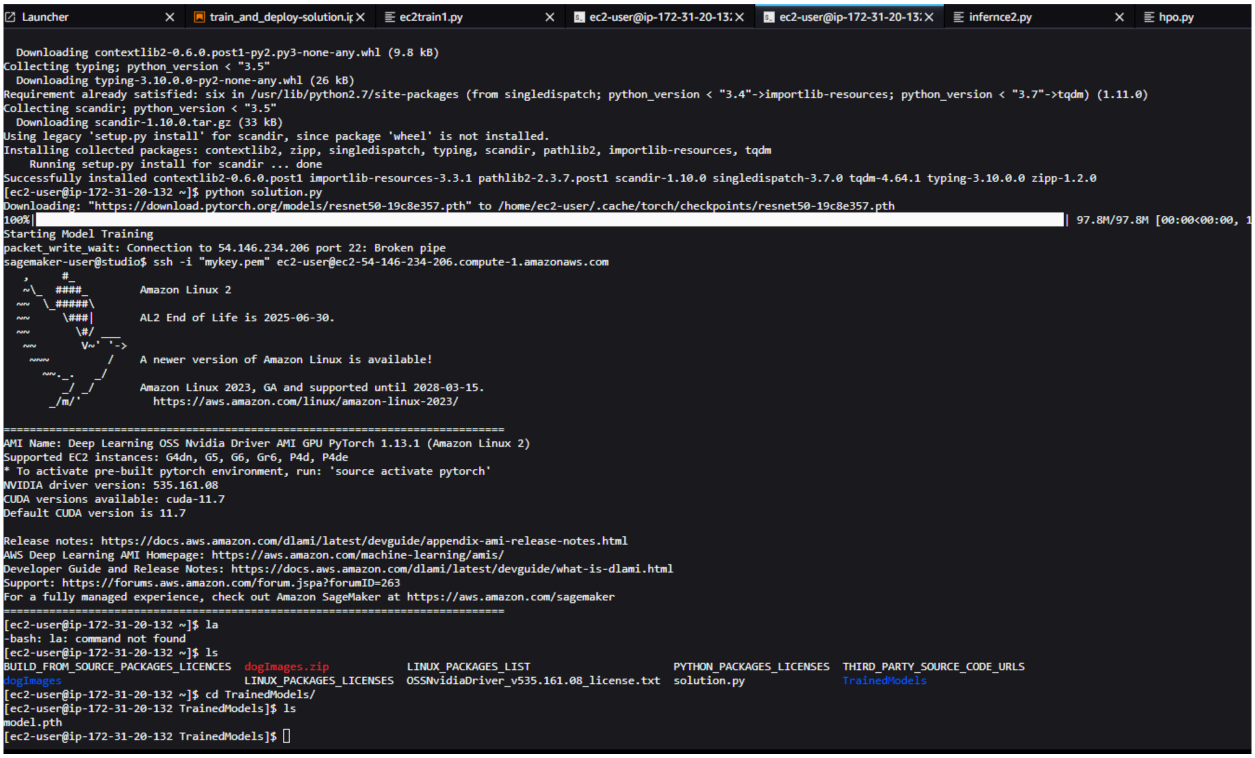
Instances (1) Info

Instance state = running

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Publ
□	udacityprojpytorch	i-08da72cce0e727dbf	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1b	ec2-35-172-221-0.com...	35.1

Write at least a paragraph about the difference between ec2train1.py and code used in step 1

This script trains the model using local system to get accessibility to data whereas sagemaker needs access to s3 , with ec2 instance it is like a local machine where you have your own storage and own computation structure, you can easily configure and install simple libraries and then run the script , i installed the numpy and torch library in the ec2 instance and then run the python solution file which was provided , after successful training of model it creates the model.pth which can be used afterwards

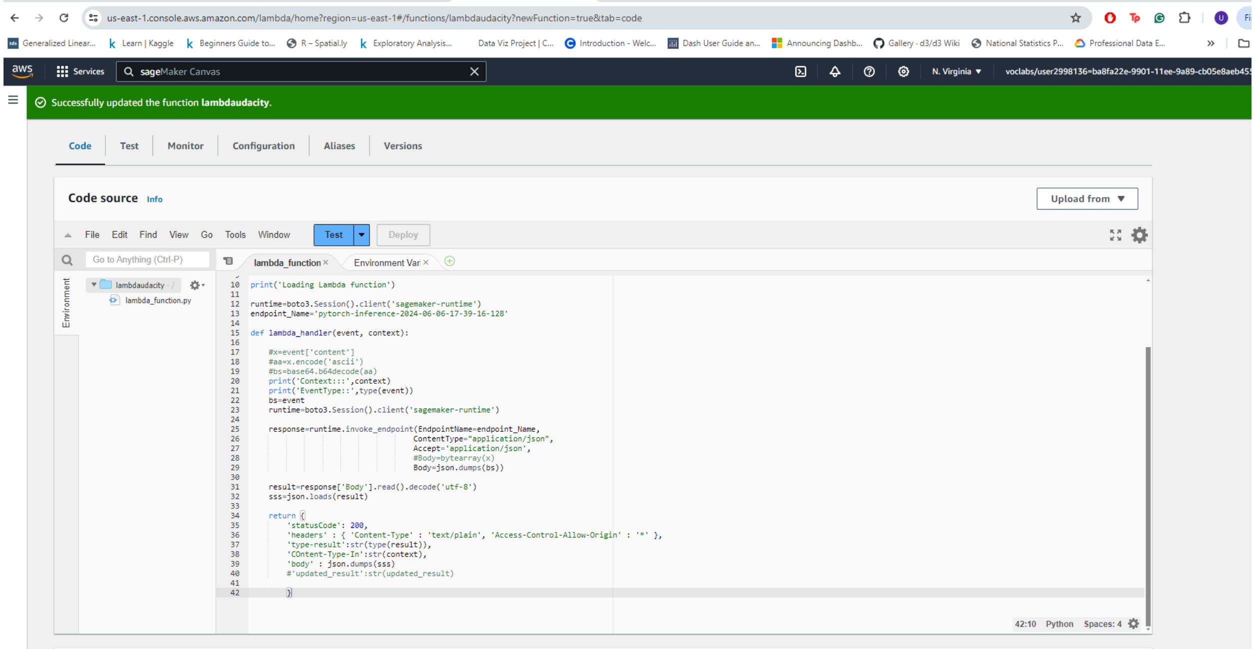

Launcher
train_and_deploy-solution.ip
ec2train1.py
ec2-user@ip-172-31-20-13:
ec2-user@ip-172-31-20-13:
infernce2.py
hpo.py

```
  Downloading contextlib2-0.6.0.post1-py2.py3-none-any.whl (9.8 kB)
Collecting typing; python_version < "3.5"
  Downloading typing-3.10.0.0-py2-none-any.whl (26 kB)
Requirement already satisfied: six in /usr/lib/python2.7/site-packages (from singledispatch; python_version < "3.4"->importlib-resources; python_version < "3.7"->tqdm) (1.11.0)
Collecting scandir; python_version < "3.5"
  Downloading scandir-1.10.0.tar.gz (33 kB)
Using legacy 'setup.py install' for scandir, since package 'wheel' is not installed.
Installing collected packages: contextlib2, zipp, singledispatch, typing, scandir, pathlib2, importlib-resources, tqdm
    Running setup.py install for scandir ... done
Successfully installed contextlib2-0.6.0.post1 importlib-resources-3.3.1 pathlib2-2.3.7.post1 scandir-1.10.0 singledispatch-3.7.0 tqdm-4.64.1 typing-3.10.0.0 zipp-1.2.0
[ec2-user@ip-172-31-20-132 ~]$ python solution.py
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /home/ec2-user/.cache/torch/checkpoints/resnet50-19c8e357.pth
100%|████████████████████████████████████████████████████████████████| 97.8M/97.8M [00:00<00:00, 1
Starting Model Training
packet_write_wait: Connection to 54.146.234.206 port 22: Broken pipe
sagemaker-user@studio$ ssh -i "mykey.pem" ec2-user@ec2-54-146-234-206.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        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/

=============================================================================
AMI Name: Deep Learning OSS Nvidia Driver AMI GPU PyTorch 1.13.1 (Amazon Linux 2)
Supported EC2 instances: G4dn, G5, G6, Gr6, P4d, P4de
* To activate pre-built pytorch environment, run: 'source activate pytorch'
NVIDIA driver version: 535.161.08
CUDA versions available: cuda-11.7
Default CUDA version is 11.7

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
=============================================================================
[ec2-user@ip-172-31-20-132 ~]$ la
-bash: la: command not found
[ec2-user@ip-172-31-20-132 ~]$ ls
BUILD_FROM_SOURCE_PACKAGES_LICENCES  dogImages.zip           LINUX_PACKAGES_LIST                        PYTHON_PACKAGES_LICENSES  THIRD_PARTY_SOURCE_CODE_URLS
dogImages                            LINUX_PACKAGES_LICENSES OSSNvidiaDriver_v535.161.08_license.txt    solution.py               TrainedModels
[ec2-user@ip-172-31-20-132 ~]$ cd TrainedModels/
[ec2-user@ip-172-31-20-132 TrainedModels]$ ls
model.pth
[ec2-user@ip-172-31-20-132 TrainedModels]$ 
```

Step 3: Lambda function setup and Step 4 attaching security to lambda function

Describe how the function is written and how it works

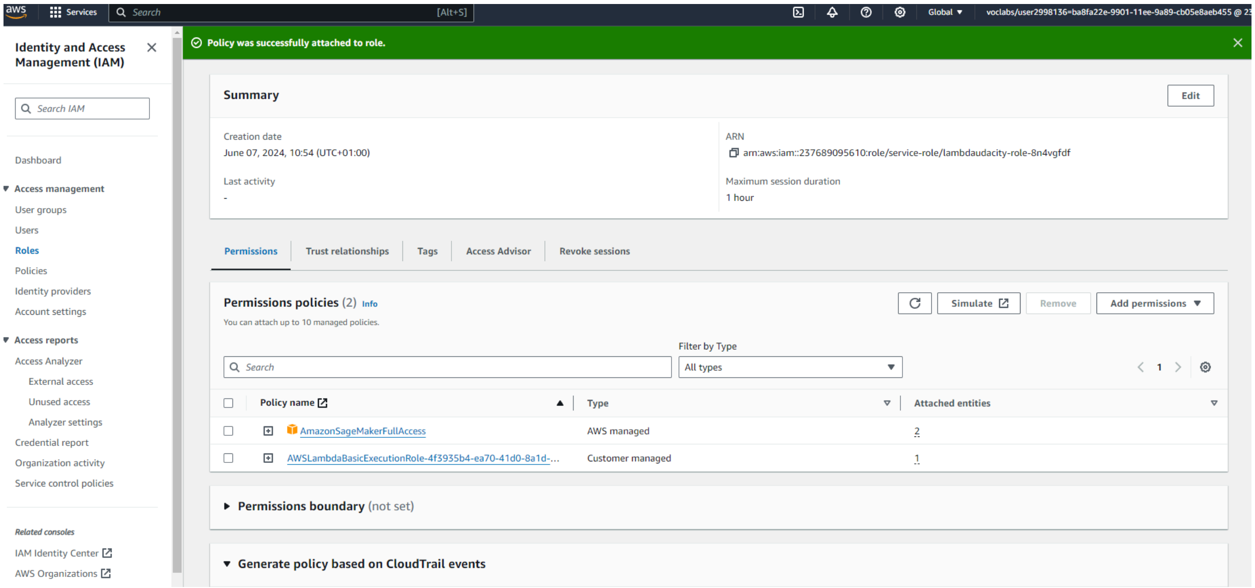
This Lambda function invokes the deployed endpoint and creates a connection with it , it sends the url provided in the test and gets a response back , but before that i had to attach the aws sagemaker full execution role policy to this lambda function , the test was successful as you can also see in the screenshot provided
Deploying and writing the lambda function



Successfully updated the function **lambdaudacity**.

```
print('Loading Lambda function')

runtime=boto3.Session().client('sagemaker-runtime')
endpoint_Name='pytorch-inference-2024-06-06-17-39-16-128'

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')

    response=runtime.invoke_endpoint(EndpointName=endpoint_Name,
                                    ContentType="application/json",
                                    Accept='application/json',
                                    #Body=bytearray(x)
                                    Body=json.dumps(bs))

    result=response['Body'].read().decode('utf-8')
    sss=json.loads(result)

    return {
        'statusCode': 200,
        'headers' : { 'Content-Type' : 'text/plain', 'Access-Control-Allow-Origin' : '*' },
        'type-result':str(type(result)),
        'COntent-Type-In':str(context),
        'body' : json.dumps(sss)
        #'updated_result':str(updated_result)

        }
```

Attaching the policy to the lambda function



Policy was successfully attached to role.

Summary

Creation date: June 07, 2024, 10:54 (UTC+01:00)

ARN: arn:aws:iam::237689095610:role/service-role/lambdaudacity-role-8n4vgfdf

Last activity: -

Maximum session duration: 1 hour

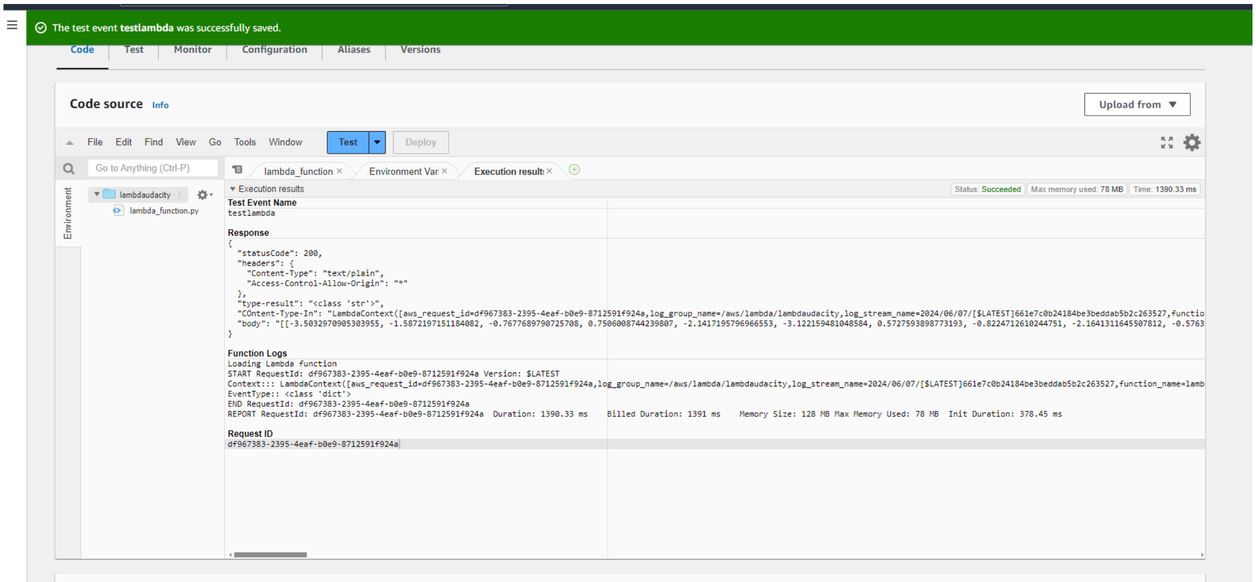
Permissions policies (2)

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonSageMakerFullAccess	AWS managed	2
AWSLambdaBasicExecutionRole-4f3935b4-ea70-41d0-8a1d-...	Customer managed	1

Checking the response back from the test function



Status: Succeeded | Max memory used: 78 MB | Time: 1390.33 ms

Test Event Name

```
testlambda
```

Response

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "LambdaContext([aws_request_id=df967383-2395-4eaf-b0e9-8712591f924a,log_group_name=/aws/lambda/lambdaudacity,log_stream_name=2024/06/07/[$LATEST]661e7c0b24184be3beddab5b2c263527,functio
  "body": "[[-3.5032970905303955, -1.5872197151184082, -0.7677689790725708, 0.7506008744239807, -2.1417195796966553, -3.122159481048584, 0.5727593898773193, -0.8224712610244751, -2.1641311645507812, -0.5763
}
```

Function Logs

```
Loading Lambda function
START RequestId: df967383-2395-4eaf-b0e9-8712591f924a Version: $LATEST
Context::: LambdaContext([aws_request_id=df967383-2395-4eaf-b0e9-8712591f924a,log_group_name=/aws/lambda/lambdaudacity,log_stream_name=2024/06/07/[$LATEST]661e7c0b24184be3beddab5b2c263527,function_name=lamb
EventType:: <class 'dict'>
END RequestId: df967383-2395-4eaf-b0e9-8712591f924a
REPORT RequestId: df967383-2395-4eaf-b0e9-8712591f924a  Duration: 1390.33 ms    Billed Duration: 1391 ms    Memory Size: 128 MB Max Memory Used: 78 MB  Init Duration: 378.45 ms
```

Request ID

```
df967383-2395-4eaf-b0e9-8712591f924a
```

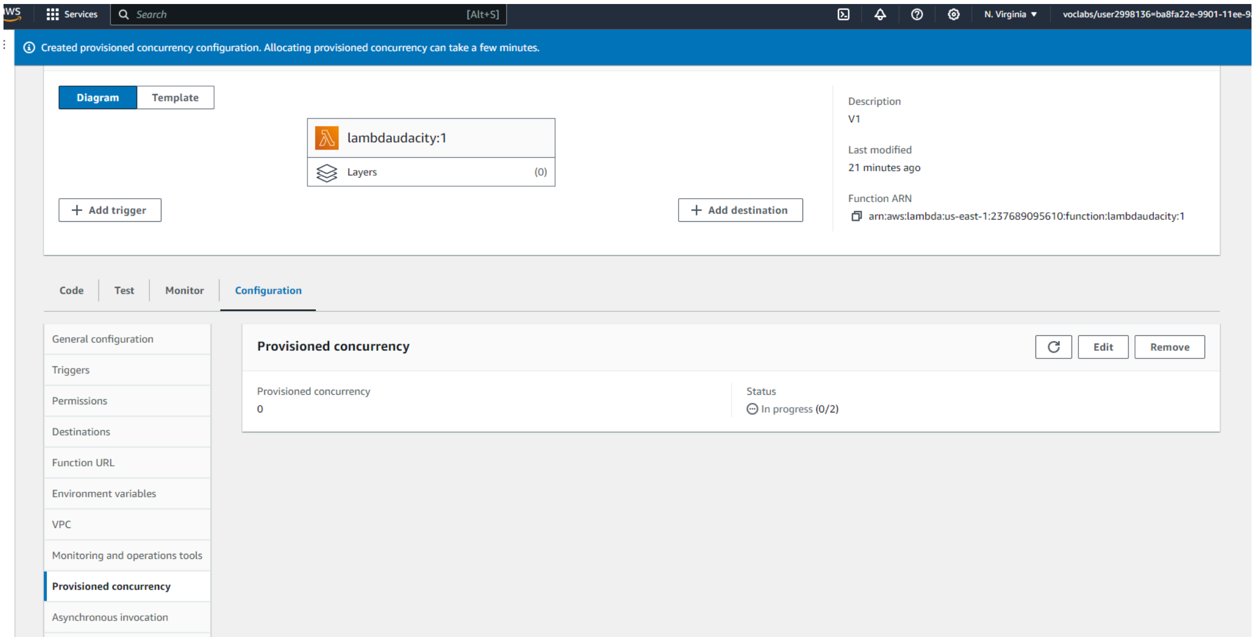
Testing my Lambda function and the response from it

[-3.5032970905303955, -1.5872197151184082, -0.7677689790725708, 0.7506008744239807, -2.1417195796966553, -3.122159481048584, 0.5727593898773193, -0.8224712610244751, -2.1641311645507812, -0.5763108730316162, -0.8086022734642029, -2.389901876449585, -1.9422714710235596, 1.100341796875, -0.26965034008026123, 0.21998783946037292, -4.448609828948975, -1.4109994173049927, -2.8092644214630127, 1.148894190788269, -1.8793400526046753, 0.2365783154964447, -4.0021281242370605, -3.8213226795196533, -2.092172384262085, -5.77842378616333, -1.631203532218933, -0.36555516719818115, -3.2245607376098633, -0.5889348387718201, -0.9163044095039368, -1.3189921379089355, -3.1707143783569336, -0.8363109827041626, -4.637266159057617, -3.038100481033325, -2.3919320106506348, -1.856014370918274, 0.7525979280471802, -2.055586814880371, -0.2888779938220978, -0.931816816329956, 1.2027690410614014, -0.7588621973991394, -2.080308675765991, -4.121342658996582, -0.23925864696502686, -0.4919027090072632, -1.1479555368423462, -0.9333341121673584, -1.6911085844039917, -2.2167131900787354, -2.755300998687744, -1.782326340675354, -2.2286133766174316, -1.389290690422058, -2.8503918647766113, -3.0832958221435547, -0.8418097496032715, 0.2474696785211563, -3.468285083770752, -3.046025276184082, -3.208275318145752, -2.329455852508545, -3.292487621307373, -2.7571160793304443, 0.29264193773269653, -2.2366702556610107, 0.6742424964904785, -0.44905605912208557, -0.5664674639701843, -2.5348920822143555, -1.873576045036316, -1.817552924156189, -2.528771162033081, -1.2557942867279053, -2.237632989883423, 0.4063810110092163, -2.647940158843994, -1.7562816143035889, 0.5188231468200684, -3.0768613815307617, -0.5471802353858948, -0.9216118454933167, -2.7565505504608154, -2.093254804611206, -1.695647954940796, -1.180632472038269, -0.8637321591377258, -0.33544591069221497, -2.9250197410583496, -3.5262932777404785, -3.9739744663238525, -2.7770373821258545, -2.3573548793792725, -1.3743940591812134, -2.9559221267700195, -2.738215446472168, -1.6649876832962036, -2.1590394973754883, -5.241294860839844, -1.4712973833084106, -0.40772566199302673, -4.421698093414307, -2.7951135635375977, -2.847907304763794, -4.077500343322754,

```
-0.6380893588066101, -2.5278351306915283, 0.6058333516120911, -1.5183559656143188,
-1.4645717144012451, -1.57234787940979, -1.371683955192566, -2.6448655128479004,
-1.3635883331298828, -3.314955949783325, -0.9221130013465881, -2.977008104324341,
-1.1008497476577759, -0.4431454539299011, -1.3449079990386963, -4.170413494110107,
-1.8815847635269165, -5.411243915557861, -2.5093414783477783, -2.258676052093506,
-1.4860609769821167, -1.7014336585998535, -1.3456860780715942, -4.2330427169799805,
-2.9709153175354004, -3.000408172607422]]"
```

Step 5 Concurrency and autoscalling

Setup of concurrency on my lambda function after defining the version



Created provisioned concurrency configuration. Allocating provisioned concurrency can take a few minutes.

Description
V1

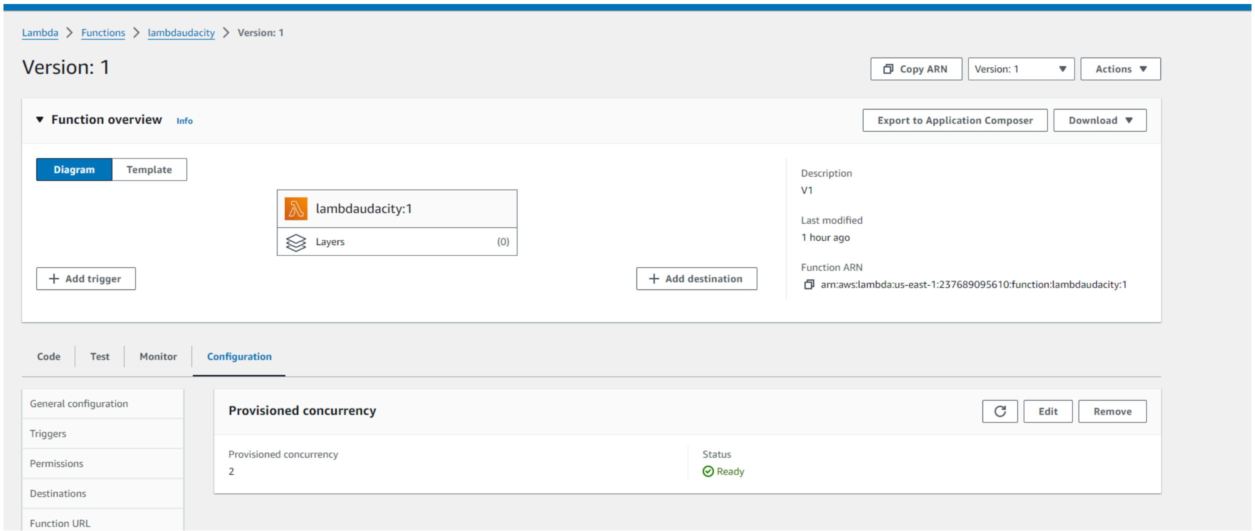
Last modified
21 minutes ago

Function ARN
arn:aws:lambda:us-east-1:237689095610:function:lambdaudacity:1

Provisioned concurrency

Provisioned concurrency: 0

Status: In progress (0/2)



Version: 1

Description
V1

Last modified
1 hour ago

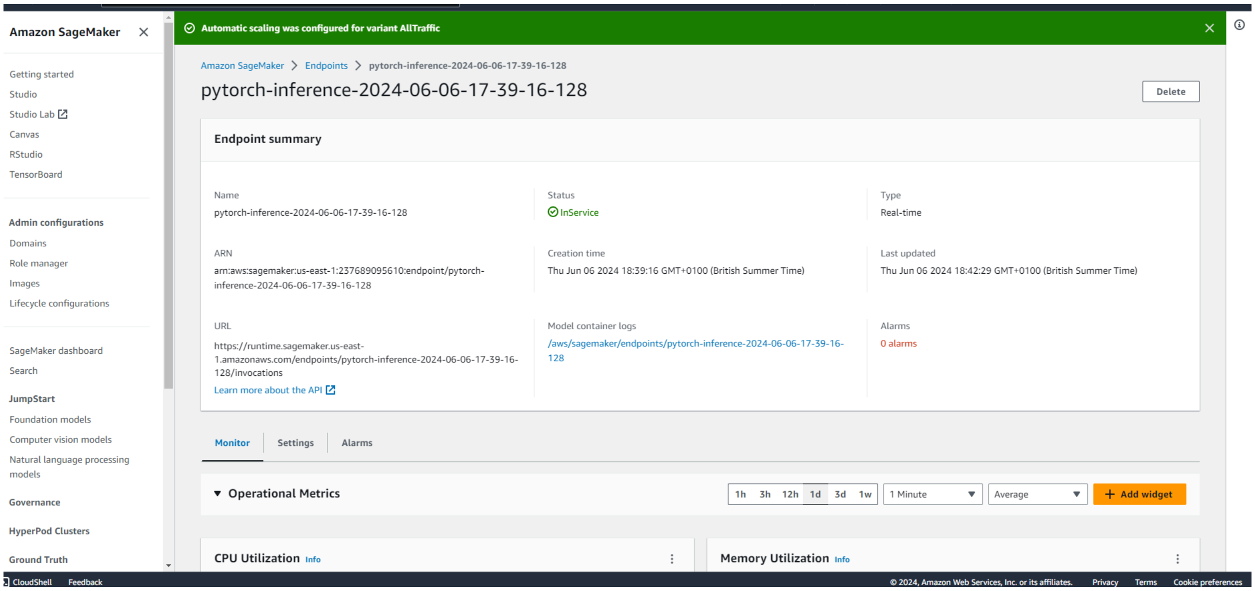
Function ARN
arn:aws:lambda:us-east-1:237689095610:function:lambdaudacity:1

Provisioned concurrency

Provisioned concurrency
2

Status
Ready

Setting autoscalling for the endpoint



Automatic scaling was configured for variant AllTraffic

pytorch-inference-2024-06-06-17-39-16-128

Endpoint summary

Name
pytorch-inference-2024-06-06-17-39-16-128

Status
InService

Type
Real-time

ARN
arn:aws:sagemaker:us-east-1:237689095610:endpoint/pytorch-inference-2024-06-06-17-39-16-128

Creation time
Thu Jun 06 2024 18:39:16 GMT+0100 (British Summer Time)

Last updated
Thu Jun 06 2024 18:42:29 GMT+0100 (British Summer Time)

URL
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2024-06-06-17-39-16-128/invocations
Learn more about the API

Model container logs
/aws/sagemaker/endpoints/pytorch-inference-2024-06-06-17-39-16-128

Alarms
0 alarms