

# Operationalizing Machine Learning on SageMaker

## Step 1 Training & Deployment

I chose the smallest SageMaker instance accessible for my notebook, 'ml.t2.medium', because I need to keep the notebook open throughout the project and don't need a very powerful instance in terms of CPU or RAM.

The screenshot shows the Amazon SageMaker console. A green banner at the top says "Success! Your notebook instance is being created. Open the notebook instance when status is InService and open a template notebook to get started." Below this, the "Notebook instances" section displays a table with one row:

Name	Instance	Creation time	Status	Actions
c4-Operational-ML-SageMaker	ml.t2.medium	Dec 30, 2021 11:30 UTC	Pending	-

The screenshot shows the Amazon S3 console. The left sidebar shows options like Buckets, Storage Lens, and AWS Marketplace. The main area shows the contents of the bucket "c4-operational-ml-sagemaker-bucket". It lists three objects/folders: "test/", "train/", and "valid/".

I have then added the address of my S3 bucket `s3://c4-operational-ml-sagemaker-bucket` to all my relevant cell in the `train_and_deploy-solution.ipynb` notebook.

Additionally, I chose to use `m1.m5.2xlarge` for both tuning and training because it has greater processing power so that the tuning job and training jobs could be completed more quickly.

To speed up tuning and ensure that better hyperparameters were chosen, I increased `max_jobs` to 10 for tuning, `max_parallel_jobs` to 3, and early stopping type to "Auto."

Amazon SageMaker

Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia v vocabs/user1681020-11709878668 @ 4583-2159-8126 ▾

Resource Groups & Tag Editor

Amazon SageMaker > Training jobs

Training jobs

Actions Create training job

Search training jobs

Name Creation time Duration Status

Name	Creation time	Duration	Status
dog-pytorch-2021-12-31-02-28-27-245	Dec 31, 2021 02:28 UTC	-	InProgress
pytorch-training-211231-0130-010-0d7a6268	Dec 31, 2021 02:05 UTC	5 minutes	Failed
pytorch-training-211231-0130-009-18f51479	Dec 31, 2021 01:59 UTC	5 minutes	Failed
pytorch-training-211231-0130-008-b94654b7	Dec 31, 2021 01:58 UTC	5 minutes	Failed
pytorch-training-211231-0130-007-250f2efb	Dec 31, 2021 01:57 UTC	22 minutes	Completed
pytorch-training-211231-0130-006-acd7cc16	Dec 31, 2021 01:53 UTC	6 minutes	Failed
pytorch-training-211231-0130-005-615494ff	Dec 31, 2021 01:36 UTC	21 minutes	Completed
pytorch-training-211231-0130-004-7a54168a	Dec 31, 2021 01:35 UTC	23 minutes	Completed
pytorch-training-211231-0130-003-fc8215db	Dec 31, 2021 01:30 UTC	5 minutes	Failed
pytorch-training-211231-0130-002-e0b63245	Dec 31, 2021 01:30 UTC	6 minutes	Failed

Feedback English (US) © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The following is my trained endpoint as per the image below

- Single instance endpoint: pytorch-inference-2021-12-31-02-52-43-385
- Multi instance endpoint: pytorch-inference-2021-12-31-03-25-17-395

Amazon SageMaker

Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia v vocabs/user1681020-11709878668 @ 4583-2159-8126 ▾

Resource Groups & Tag Editor

Amazon SageMaker > Endpoints

Endpoints

Update endpoint Actions Create endpoint

Search endpoints

Name ARN Creation time Status Last updated

Name	ARN	Creation time	Status	Last updated
pytorch-inference-2021-12-31-03-25-17-395	arn:aws:sagemaker:us-east-1:458321598126:endpoint/pytorch-inference-2021-12-31-03-25-17-395	Dec 31, 2021 03:25 UTC	InService	Dec 31, 2021 03:28 UTC
pytorch-inference-2021-12-31-02-52-43-385	arn:aws:sagemaker:us-east-1:458321598126:endpoint/pytorch-inference-2021-12-31-02-52-43-385	Dec 31, 2021 02:52 UTC	InService	Dec 31, 2021 02:55 UTC

Feedback English (US) © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Here I have used the image Deep Learning AMI (Amazon Linux 2) Version 56.0 and the instance type 't3.xlarge'. This seemed to me to be a nice balance of performance and affordability. Because we are running a spot instance EC2 for training, it is logical to choose a more powerful instance type than is required so that we are able to quickly train our model, and the cost for spot instance are quite small. I have also limited the access to my EC2 instance to only my IP address for improved security.

**Step 2: Choose an Instance Type**

Name	Instance Type	vCPUs	Memory (GiB)	Root device type	Network	Support	Price
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	<b>t2.micro</b> <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<b>t3</b>	<b>t3.large</b>	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
t3a	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
t3a	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
t3a	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

**Feedback English (US) ▾** © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

  

**Instances (1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP
i-0193f6abb6c7adb8	Running	t3.large	Initializing	No alarms	+ us-east-1c	ec2-54-227-63-165.co...	54.227.63.165	

Select an instance

**Feedback English (US) ▾** © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group												
Purchasing option	<input checked="" type="checkbox"/> Request Spot instances													
Current price	<table border="1"> <thead> <tr> <th>Availability Zone</th> <th>Current price</th> </tr> </thead> <tbody> <tr><td>us-east-1a</td><td>\$0.0268</td></tr> <tr><td>us-east-1b</td><td>\$0.0282</td></tr> <tr><td>us-east-1c</td><td>\$0.0266</td></tr> <tr><td>us-east-1d</td><td>\$0.027</td></tr> <tr><td>us-east-1f</td><td>\$0.0275</td></tr> </tbody> </table>		Availability Zone	Current price	us-east-1a	\$0.0268	us-east-1b	\$0.0282	us-east-1c	\$0.0266	us-east-1d	\$0.027	us-east-1f	\$0.0275
Availability Zone	Current price													
us-east-1a	\$0.0268													
us-east-1b	\$0.0282													
us-east-1c	\$0.0266													
us-east-1d	\$0.027													
us-east-1f	\$0.0275													
Maximum price	\$ [e.g. 0.045 = 4.5 cents/hour (Optional)]													
Persistent request	<input type="checkbox"/> Persistent request													
Network	vpc-040ddd1f94ec09120c (default)	<input type="button" value="Create new VPC"/>												
Subnet	No preference (default subnet in any Availability Zone)	<input type="button" value="Create new subnet"/>												
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)													
Hostname type	<input type="checkbox"/> Use subnet setting (IP name)													
DNS Hostname	<input type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests													
Placement group	<input type="checkbox"/> Add instance to placement group													
Capacity Reservation	<input type="button" value="Open"/>													

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

**Step 1: Choose an Amazon Machine Image (AMI)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our partners, or the community.

Deep Learning AMI (Amazon Linux 2)

Quick Start (1)	<b>Deep Learning AMI (Amazon Linux 2) Version 56.0 - ami-0b331a9baeb8467ca</b> MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.3, 2.3.4 & 1.15.5, PyTorch-1.7.1 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVL Amazon Linux Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
The following results for "Deep Learning AMI (Amazon Linux 2)" were found in other catalogs: <ul style="list-style-type: none"> <li>14 results in AWS Marketplace</li> <li>AWS Marketplace provides partnered Software that is pre-configured to run on AWS</li> <li>120 results in Community AMIs</li> <li>Community AMIs are AMIs that are shared by the general AWS community</li> </ul>	

```

model.eval()
running_loss=0
running_corrects=0

for inputs, labels in test_loader:
    outputs=model(inputs)
    loss=criterion(outputs, labels)
    _, preds = torch.max(outputs, 1)
    running_loss += loss.item() * inputs.size(0)
    running_corrects += torch.sum(preds == labels.data)

total_loss = running_loss / len(test_loader)
total_acc = running_corrects.double() / len(test_loader)

def train(model, train_loader, validation_loader, criterion, optimizer):
    epochs=5
    best_loss=1e6
    image_dataset={'train':train_loader, 'valid':validation_loader}
"solution.py" 147L, 4843B written
[root@ip-172-31-22-94 ~]# python solution.py
Traceback (most recent call last):
  File "solution.py", line 1, in <module>
    import numpy as np
ImportError: No module named numpy
[root@ip-172-31-22-94 ~]# source activate pytorch_latest_p37
(pytorch_latest_p37) [root@ip-172-31-22-94 ~]# fit
-bash: fit: command not found
(pytorch_latest_p37) [root@ip-172-31-22-94 ~]# dir
dogImages dogImages.zip solution.py TrainedModels
(pytorch_latest_p37) [root@ip-172-31-22-94 ~]# ls
dogImages dogImages.zip solution.py TrainedModels
(pytorch_latest_p37) [root@ip-172-31-22-94 ~]# python solution.py
^[[A
ls
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /root/.cache/torch/hub/checkpoints/res
100%|██████████| 2.11M/2.11M
Starting Model Training

```

i-0ac1c27ffccaa11ca

Public IPs: 54.167.105.124 Private IPs: 172.31.22.94

screenshot showing the result of model.pth

```

Deep Learning AMI (Amazon Linux) Version 56.0
=====
Please use one of the following commands to start the required environment with the framework of your choice:
for AWS MX 1.7 (+Keras2) with Python3 (CUDA 10.1 and Intel MKL-DNN) _____ source activate mxnet_p36
for AWS MX 1.8 (+Keras2) with Python3 (CUDA + and Intel MKL-DNN) _____ source activate mxnet_latest_p37
for AWS MX (+AWS Neuron) with Python3 _____ source activate aws_neuron_mxnet_p36
for AWS MX (+Amazon Elastic Inference) with Python3 _____ source activate amazonei_mxnet_p36
for TensorFlow(+Keras2) with Python3 (CUDA + and Intel MKL-DNN) _____ source activate tensorflow_p37
for TensorFlow(+AWS Neuron) with Python3 _____ source activate aws_neuron_tensorflow_p36
for TensorFlow(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_tensorflow_p36
for TensorFlow 2(+Keras2) with Python3.7 (CUDA + and Intel MKL-DNN) _____ source activate tensorflow2_p37
for TensorFlow 2.4 with Python3.7 (CUDA + and Intel MKL-DNN) _____ source activate tensorflow2_latest_p37
for TensorFlow 2(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_tensorflow2_p36
for PyTorch 1.7.1 with Python3.7 (CUDA 11.0 and Intel MKL) _____ source activate pytorch_p37
for PyTorch 1.8.1 with Python3.7 (CUDA 11.1 and Intel MKL) _____ source activate pytorch_latest_p37
for PyTorch (+AWS Neuron) with Python3 _____ source activate aws_neuron_pytorch_p36
for PyTorch 1.5.1 (+Amazon Elastic Inference) with Python3 _____ source activate amazonei_pytorch_latest_p37
for base Python3 (CUDA 10.0) _____ source activate python3

To automatically activate base conda environment upon login, run: 'conda config --set auto_activate_base true'

Official Conda User Guide: https://docs.conda.io/projects/conda/en/latest/user-guide/
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
When using INF1 type instances, please update regularly using the instructions at: https://github.com/aws/aws-neuron-sdk/tree/master/release-notes
Security scan reports for python packages are located at: /opt/aws/dlami/info/
=====
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[root@ip-172-31-22-94 ~]# ls
dogImages dogImages.zip solution.py TrainedModels
[root@ip-172-31-22-94 ~]# cd TrainedModels/
[root@ip-172-31-22-94 TrainedModels]# ls
model.pth
[root@ip-172-31-22-94 TrainedModels]# 
```

### Differences between ec2train1.py and hpo.py

When comparing the EC2 code to Step 1. The code that we worked in Step 1 was different as the code was written to work with SageMaker, and there are some differences in the code that we used in etrain1.py. The argument parsing at the main are only done in Sage maker to print the argument out to notebook, which we don't do when training in EC2. The modules that we used are only can be used in SageMaker. such as Logger.info that are used in SageMaker. Model debug hook for cloudwatch is another example as well that not in this example. I do also notice that the ec2 training using testdata.

## Step 3 Lambda functions

I have created the lambda function to the first endpoint I created to test the single instance trained endpoint. We are provided lamdafunction.py here and I have changed added the appropriate single instance endpoint name here `endpoint\_Name='pytorch-inference-2021-12-31-02-52-43-385'` Since we are provided as a lamdafunction.py file, I have decided to just upload it as a zip. This lambda function essentially just invoke our SageMaker endpoint using the boto3 API from SageMaker. The response from the endpoint that was invoked, will then be read and be utf-8 decoded. The results are then converted to a JSON and return on the Body element. I also noticed that the runtime here is also declared twice. The lambda function take in the following Test Json as the input which is the url of an image of a dog. { "url": "https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2017/11/20113314/Carolina-Dog-standing-outdoors.jpg" }

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with tabs for 'Functions - Lambda', 'Services', and a search bar. Below the navigation bar, the main content area has a heading 'Lambda > Functions'. A message box displays the warning: '⚠ Tags failed to load. The filter doesn't include tags.' Below this, a table lists one function:

Function name	Description	Package type	Runtime	Code size	Last modified
operationalmltestendpoint	-	Zip	Python 3.8	743.0 byte	26 seconds ago

At the bottom of the page, there are links for 'Feedback', 'English (US)', '© 2021, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

## Step 4 Security & Testing

In any Role-based access control, permission should always be given the least privilege, but in this example here, we have given the full access to the SageMaker. We could identify which service that the endpoint is using and we can just give that particular access. We could also introduce Session Authentication on the lambda function which only allow authenticated users to call the endpoints. We may also need to review the roles to make sure that all the roles that we are using are current and not inactive, to reduce the risk that it might belong to previous colleagues. The root users that we used in this example also didn't utilize MFA as per the warning that we got on the IAM dashboard.

I got the following output before adding full access to SageMaker to the role running the lambda functions

operationalmltestendpoint - Lambda

console.aws.amazon.com/lambda/home?region=us-east-1#/functions/operationalmltestendpoint?tab=code

Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia v vclabs/user1681020-11709878668 @ 4583-2159-8126

Resource Groups & Tag Editor

The test event **test-dog** was successfully saved.

**Function overview** Info

**operationalmltest endpoint**

Layers (0)

+ Add trigger + Add destination

Description -

Last modified 5 minutes ago

Function ARN arn:aws:lambda:us-east-1:458321598126:function:operationalmltestendpoint

Code Test Monitor Configuration Aliases Versions

**Code source** Info

File Edit Find View Go Tools Window Test Deploy Changes deployed

Go to Anything (Ctrl-P)

Environment operationalmltest endpoint lambda\_function.py

Execution results Test Event Name test-dog Status: Failed Max memory used: 65 MB Time: 738.03 ms

Response

```
{
  "errorMessage": "An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::458321598126:SessionId:1cd850ac-ab5a-4206-aa0b-87c6f8d748d5, Action: invoke, Resource: arn:aws:lambda:us-east-1:458321598126:function:operationalmltestendpoint, Region: us-east-1, ResponseMetadata: {RequestId: 1cd850ac-ab5a-4206-aa0b-87c6f8d748d5, StatusCode: 403, HttpStatusCode: 403, Headers: {Content-Type: application/json}, Duration: 738.03ms, RequestId: 1cd850ac-ab5a-4206-aa0b-87c6f8d748d5, HostId: 1cd850ac-ab5a-4206-aa0b-87c6f8d748d5, Status: OK}, 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 386, in _api_call\n        return self._make_api_call(operation_name, kwargs)\n    \"File \"\\"/var/runtime/botocore/client.py\\\"", line 705, in _make_api_call\n        raise error_class(parsed_response, operation_name)\n"
  ]
}
```

Function Logs START RequestId: c1d850ac-ab5a-4206-aa0b-87c6f8d748d5 Version: \$LATEST Loading Lambda function

Feedback English (US) © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

IAM Management Console

console.aws.amazon.com/iam/home?#roles/operationalmltestendpoint-role-33bp10xw?section=permissions

Services Search for services, features, blogs, docs, and more [Alt+S] Global v vclabs/user1681020-11709878668 @ 4583-2159-8126

Resource Groups & Tag Editor

Identity and Access Management (IAM)

Dashboard

Access management User groups Users

**Roles**

Policies Identity providers Account settings

Access reports Access analyzer Archive rules Analyzers Settings

Credential report Organization activity Service control policies (SCPs)

Search IAM

AWS account ID: 458321598126

New feature to generate a policy based on CloudTrail events. AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this role.

Roles > operationalmltestendpoint-role-33bp10xw Summary Delete role

Role ARN arn:aws:iam::458321598126:role/service-role/operationalmltestendpoint-role-33bp10xw

Role description Edit

Instance Profile ARNs

Path /service-role/

Creation time 2021-12-31 16:13 UTC+1100

Last activity Not accessed in the tracking period

Maximum session duration 1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (2 policies applied)

Attach policies Add inline policy

Policy name	Policy type
AmazonSageMakerFullAccess	AWS managed policy
AWSLambdaBasicExecutionRole-6b0ac66a-2fc8-4844-...	Managed policy

Permissions boundary (not set)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. Learn more

Set boundary

Feedback English (US) © 2021, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

I got the following output from a test after attaching full access role to SageMaker on the role running the lambda function:

The screenshot shows the AWS Lambda console interface. At the top, there's a header with the URL "console.aws.amazon.com/lambda/home?region=us-east-1#/functions/operationalmltestendpoint?tab=code". Below the header, there's a search bar and a "Resource Groups & Tag Editor" button. The main area has tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions". The "Code" tab is active, showing the "Code source" section. It includes a file browser with "lambda\_function.py" selected, and a terminal window showing the command "lambda\_function". Below the file browser, there's a "Test" tab with dropdown menus for "Execution results", "Test Event Name", and "Test dog". The "Execution result" panel shows a green status bar with "Status: Succeeded", "Max memory used: 69 MB", and "Time: 835.68 ms". The "Response" section displays a JSON object with fields like "statusCode", "headers", and "body". The "Function Logs" section shows log entries with details like RequestId, Duration, and Memory Size. At the bottom, there are links for "Feedback", "English (US)", and "Cookie preferences".

The screenshot shows the IAM Management Console interface. The URL is "console.aws.amazon.com/iamv2/home?#/home". The left sidebar has sections for "Identity and Access Management (IAM)" (Dashboard, Access management, Access reports), "AWS CloudTrail", "AWS CloudWatch Metrics", "AWS CloudWatch Metrics Insights", "AWS CloudWatch Metrics Insights Analytics", "AWS CloudWatch Metrics Insights Metrics", and "AWS CloudWatch Metrics Insights Metrics Insights". The main content area has a blue banner saying "Introducing the new IAM dashboard experience" and "We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.". Below the banner is the "IAM dashboard" section with a table showing resource counts: User groups (0), Users (0), Roles (17), Policies (6), and Identity providers (0). There's also a "Security recommendations" section with a warning about adding MFA to the root user. The "What's new" section lists recent updates from the IAM blog. On the right, there's an "AWS Account" sidebar with fields for Account ID (458321598126), Account Alias (458321598126), and Sign-in URL (https://458321598126.ws.amazon.com/console). A "Tools" sidebar includes a "Policy simulator" and "Web identity federation" section.

## Step 5 Concurrency and auto-scaling

I chose reserved concurrency over shared concurrency because it is free and satisfies our current needs. Furthermore, we are unlikely to handle more than a few requests per endpoint instance at any given time, as this would indicate that the latter is overloaded, so having this value be a multiple of the number of endpoint instances makes sense, so I chose 100, allowing for 20 requests per endpoint instance. I decided to let the endpoint scale out to between one and five instances. It should be enough for our use in this case for this project as because each prediction only take roughly 0.835 seconds to run.

Configure provisioned concurrency

Lambda > Functions > operationalmltestendpoint > Configure provisioned concurrency

## Provisioned concurrency

Qualifier type  
You can configure provisioned concurrency for an alias or a version.

Alias  
 Version

Version  
Provision concurrency for a version.

1
Aliases:-
Description: first_version

Provisioned concurrency  
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](#)

\$6.98 per month in addition to pricing for duration and requests. [Pricing](#)

5
---

100 available

Cancel **Save**

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

on:operationalmltestendpoint

Code Test Monitor Configuration Aliases Versions

### Concurrency

Function concurrency	Reserved concurrency
Use reserved concurrency	100

### Provisioned concurrency configurations (1)

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
1	version	0	In progress (0/5)	-

Amazon SageMaker

console.aws.amazon.com/sagemaker/home?region=us-east-1#endpoints/pytorch-inference-2021-12-31-02-52-43-385/autoscaling/AllTraffic

Resource Groups & Tag Editor

### Configure variant automatic scaling

Deregister auto scaling

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

Minimum instance count: 1 Maximum instance count: 5

IAM role: Amazon SageMaker uses the following service-linked role for automatic scaling. [Learn more](#)  
AWSLambdaRoleForApplicationAutoScaling\_SageMakerEndpoint

### Built-in scaling policy

Policy name: SageMakerEndpointInvocationScalingPolicy

Target metric	Target value
SageMakerVariantInvocationsPerInstance	5

Scale in cool down (seconds) - optional: 300 Scale out cool down (seconds) - optional: 300

Disable scale in: Select if you don't want automatic scaling to delete instances when traffic decreases. [Learn more](#)

Amazon SageMaker

console.aws.amazon.com/sagemaker/home?region=us-east-1#endpoints/pytorch-inference-2021-12-31-02-52-43-385

Resource Groups & Tag Editor

### Monitor

Access CloudWatch logs to view your Jupyter notebook's debugging and progress reporting. [Learn more](#)

View invocation metrics View instance metrics View logs

Endpoint runtime settings

Variant name	Current weight	Desired weight	Instance type	Elastic Inference	Current instance count	Desired instance count	Instance min - max	Automatic scaling
AllTraffic	1	1	mL.m5.large	-	1	1	1 - 5	Yes

Endpoint configuration settings

Endpoint configuration

Name	ARN	Encryption key	Creation time
pytorch-inference-2021-12-31-02-52-43-385	arn:aws:sagemaker:us-east-1:458321598126:endpoint-config/pytorch-inference-2021-12-31-02-52-43-385	-	Dec 31, 2021 02:52 UTC