



Operationalizing an AWS ML Project


SageMaker instance:

I have chosen an ml.t3.medium instance because it has sufficient computing power (vCPU =2, Memory = 4 GiB) and it's cheap (Price per Hour = \$0.05). So, it does the job while maintaining a low cost.

 **Success! Your notebook instance is being created.**
Open the notebook instance when status is InService and open a template notebook to get started.

[View details](#) 

[Amazon SageMaker](#) > [Notebook instances](#)

Notebook instances  [Actions](#) [Create notebook instance](#)

☐

Name

☐

Instance

☐

Creation time

☐

Status

☐

Actions

☐

ML-OPs-Project

☐

ml.t3.medium

☐

Nov 26, 2022 09:37 UTC

☒

InService

☐

[Open Jupyter](#) | [Open JupyterLab](#)


S3 Bucket:

[Amazon S3](#) > [Buckets](#) > [udacity40](#)

udacity40 [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (5)
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](#)

 [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

☐

Name

☐

Type

☐

Last modified

☐

Size

☐

Storage class

☐

[model/](#)

Folder

-

-

-

☐

[output/](#)

Folder

-

-

-

☐

[test/](#)

Folder

-

-

-

☐

[train/](#)

Folder

-

-

-

☐

[valid/](#)

Folder

-

-

-

Training and Deployment:

I have had several problems when training the model using the provided instance, first with the error “limit exceeded” and another because some problems in hpo.py. But I managed to overcome these problems by using “ml.m5.2xlarge” instance and max_jobs=6, max_parallel_jobs=3. The hyper-parameter and training jobs are show in the below pictures.

The image displays two screenshots of the Amazon SageMaker console, showing the results of hyperparameter tuning and training jobs.

Hyperparameter tuning jobs:

Name	Status	Training completed/total	Creation time	Duration
pytorch-training-221127-2239	Completed	6 / 6	Nov 27, 2022 22:39 UTC	27 minutes
pytorch-training-221127-2123	Failed	0 / 2 2 Failed	Nov 27, 2022 21:23 UTC	an hour
pytorch-training-221126-1126	Failed	0 / 2 2 Failed	Nov 26, 2022 11:26 UTC	6 minutes
pytorch-training-221126-1110	Failed	0 / 2 2 Failed	Nov 26, 2022 11:10 UTC	4 minutes
pytorch-training-221126-1059	Failed	0 / 2 2 Failed	Nov 26, 2022 10:59 UTC	5 minutes
pytorch-training-221126-1050	Failed	0 / 2 2 Failed	Nov 26, 2022 10:50 UTC	5 minutes

Training jobs:

Name	Creation time	Duration	Job status	Warm pool status
pytorch-training-221127-2239-006-e45a43b9	Nov 27, 2022 22:54 UTC	12 minutes	Completed	Available
pytorch-training-221127-2239-005-2d3653eb	Nov 27, 2022 22:53 UTC	12 minutes	Completed	Available
pytorch-training-221127-2239-004-bc7c234a	Nov 27, 2022 22:53 UTC	11 minutes	Completed	Available
pytorch-training-221127-2239-003-938dd3fc	Nov 27, 2022 22:39 UTC	13 minutes	Completed	Reused
pytorch-training-221127-2239-002-c4b0d24f	Nov 27, 2022 22:39 UTC	13 minutes	Completed	Reused
pytorch-training-221127-2239-001-a6df72a9	Nov 27, 2022 22:39 UTC	13 minutes	Completed	Reused
pytorch-training-221127-2123-002-93c56081	Nov 27, 2022 21:58 UTC	31 minutes	Failed	Terminated

For the training I started training using one instance and another time using multi-instance.

Amazon SageMaker > Training jobs > dog-pytorch-2022-11-27-23-21-41-654

dog-pytorch-2022-11-27-23-21-41-654

CloneCreate model packageStopCreate model

Job settings

Job name
dog-pytorch-2022-11-27-23-21-41-654

ARN
arn:aws:sagemaker:us-east-1:873635364805:training-job/dog-pytorch-2022-11-27-23-21-41-654

Status
Completed
View history

Creation time
Nov 27, 2022 23:21 UTC

Last modified time
Nov 27, 2022 23:40 UTC

SageMaker metrics time series
Enabled

Training time (seconds)
1077

Billable time (seconds)
1077

Managed spot training savings
0%

Tuning job source/parent
-

IAM role ARN
arn:aws:iam::873635364805:role/Adel-Udacity-role

Algorithm

Algorithm ARN
-

Additional volume size (GB)
30

Maximum wait time for managed spot training(s)
-

Volume encryption key
-

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

Maximum runtime (s)
86400

Managed spot training
Disabled

Input mode
File

Instance group

Instance type
ml.m5.xlarge

Instance count
1

Keep alive period
-

Amazon SageMaker > Training jobs > dog-pytorch-2022-11-27-23-36-24-463

dog-pytorch-2022-11-27-23-36-24-463

CloneCreate model packageStopCreate model

Job settings

Job name
dog-pytorch-2022-11-27-23-36-24-463

ARN
arn:aws:sagemaker:us-east-1:873635364805:training-job/dog-pytorch-2022-11-27-23-36-24-463

Status
Completed
View history

Creation time
Nov 27, 2022 23:36 UTC

Last modified time
Nov 27, 2022 23:57 UTC

SageMaker metrics time series
Enabled

Training time (seconds)
1126

Billable time (seconds)
1126

Managed spot training savings
0%

Tuning job source/parent
-

IAM role ARN
arn:aws:iam::873635364805:role/Adel-Udacity-role

Algorithm

Algorithm ARN
-

Additional volume size (GB)
30

Maximum wait time for managed spot training(s)
-

Volume encryption key
-

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

Maximum runtime (s)
86400

Managed spot training
Disabled

Input mode
File

Instance group

Instance type
ml.m5.xlarge

Instance count
5

Keep alive period
-

And here is the deployed end points (one using training with one instance and the other with multi-instance)

The screenshot shows the Amazon SageMaker console interface. The left sidebar contains navigation links for 'Getting started', 'Control panel' (Studio, Studio Lab, Canvas, RStudio), 'SageMaker dashboard' (Images, Lifecycle configurations, Search), and 'JumpStart' (Foundation models). The main content area is titled 'Endpoints' and features a search bar, a table of endpoints, and buttons for 'Update endpoint', 'Actions', and 'Create endpoint'. The table lists two endpoints, both in 'InService' status.

Name	ARN	Creation time	Status	Last updated
pytorch-inference-2022-11-28-00-09-10-211	arn:aws:sagemaker:us-east-1:873635364805:endpoint/pytorch-inference-2022-11-28-00-09-10-211	Nov 28, 2022 00:09 UTC	InService	Nov 28, 2022 00:11 UTC
pytorch-inference-2022-11-28-00-01-49-423	arn:aws:sagemaker:us-east-1:873635364805:endpoint/pytorch-inference-2022-11-28-00-01-49-423	Nov 28, 2022 00:01 UTC	InService	Nov 28, 2022 00:04 UTC

EC2:

I have used a t2.micro instance because it's eligible for the free tier and provide the needed computation power.

The screenshot shows the AWS Management Console 'Instances' page. It displays a single EC2 instance named 'EC2-for-mlops' with ID 'i-0efdc18aab027d1d'. The instance is in a 'Running' state, using a 't2.micro' instance type, and is located in the 'us-east-1d' availability zone. The status check shows '2/2 checks passed'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
EC2-for-mlops	i-0efdc18aab027d1d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-54-210-144-168.co...	54.210.144.168	-

Difference between EC2 training code and SageMaker's:

- In EC2 code there is no calling for any Estimator or Tuner functions. 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 code, all the variables already mentioned in the code itself.
- In the EC2 the training happens on the same while in the SageMaker the training job runs on a separate container than the one on which the SageMaker notebook is running.

Lambda function:

Lambda functions are used for invoking our deployed endpoints. And the end point used is the one of the mult-instance training (pytorch-inference-2022-11-28-18-38-11-439). Also we attached an Amazon SageMaker full Accesses policy to be able to interact with SageMaker successfully with no errors.

In the following images there are the role and result of the testing (also included in a separate file called lambda.txt)

The top screenshot displays the AWS IAM console for the role **mylambda-func-role-v5fpysbz**. The role was created on November 28, 2022, at 18:41 (UTC+02:00). It has an ARN of `arn:aws:iam::873635364805:role/service-role/mylambda-func-role-v5fpysbz and a maximum session duration of 1 hour. The role is associated with two permissions policies: AWSLambdaBasicExecutionRole-c4ad7a08-1ea8-436a-9cb6-5fc72705cfea (Customer managed) and AmazonSageMakerFullAccess (AWS managed). The permissions boundary is not set.`

The bottom screenshot shows the AWS Lambda console for the function **mylambda-func**. The function was successfully updated. The code source is a Python file named `lambda_function.py`. The execution results show a successful invocation with a status of `Succeeded`, a response of `200`, and a duration of `1056.84 ms`. The function logs show the following details:

```
Function Logs
Loading Lambda function
START RequestId: fdfab81c1-4ded-4f47-92b2-61106e7fe151 Version: $LATEST
Context: LambdaContext([aws_request_id=fdfab81c1-4ded-4f47-92b2-61106e7fe151, log_group_name=/aws/lambda/mylambda-func, log_stream_name=2022/11/28/[LATEST]abe445861a1f433fb57e161238da3b2, function_name=mylambda-func])
END RequestId: fdfab81c1-4ded-4f47-92b2-61106e7fe151
REPORT RequestId: fdfab81c1-4ded-4f47-92b2-61106e7fe151 Duration: 1056.84 ms Billed Duration: 1057 ms Memory Size: 128 MB Max Memory Used: 68 MB Init Duration: 323.06 ms
```

Security:

It's not a good idea to give full access permission because it may result in a security breach, but always choose the right policies and permission and remove them when the task is done. Below is the policies attached to my SageMaker execution role.

The screenshot displays the AWS IAM console interface for the 'Adel-Udacity-role'. The left sidebar shows the 'Identity and Access Management (IAM)' menu with options like 'Users', 'Groups', 'Roles', 'Policies', and 'Access reports'. The main content area shows the role's details, including its creation date (November 27, 2022, 21:51 UTC+02:00) and its ARN (arn:aws:iam:873635364805:role/Adel-Udacity-role). The 'Permissions' tab is selected, showing a list of attached policies. The table lists three AWS managed policies: 'AmazonS3FullAccess', 'AmazonSageMakerFullAccess', and 'AmazonSageMakerCanvasFullAccess'. The 'Permissions boundary' section indicates that no boundary is currently set.

Adel-Udacity-role

Allows SageMaker notebook instances, training jobs, and models to access S3, ECR, and CloudWatch on your behalf.

Summary

Creation date: November 27, 2022, 21:51 (UTC+02:00)
Last activity: 53 minutes ago
ARN: arn:aws:iam:873635364805:role/Adel-Udacity-role
Maximum session duration: 1 hour

Permissions policies (3)

You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
AmazonSageMakerFullAccess	AWS managed	Provides full access to Amazon SageM...
AmazonSageMakerCanvasFullAccess	AWS managed	Provides full access to Amazon SageM...

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

Concurrency and Auto Scaling:

I have configured a provisioned concurrency after publishing a version for lambda. Also I have configured Auto Scaling to cope with the traffic requests.

The image displays two screenshots from the AWS Management Console. The top screenshot shows the Amazon SageMaker console for an endpoint named 'pytorch-inference-2022-11-28-18-38-11-439'. A green banner at the top indicates 'Automatic scaling was configured for variant AllTraffic'. The 'Endpoint settings' section shows the endpoint is 'InService' and provides details like ARN, creation time, and URL. The bottom screenshot shows the AWS Lambda console for a function named 'mylambda-func'. The 'Configuration' tab is active, showing 'Function concurrency' set to 'Use unreserved account concurrency' and 'Unreserved account concurrency' set to 995. Under 'Provisioned concurrency configurations (1)', there is one configuration with a qualifier of '1', type of 'version', and a provisioned concurrency of 0, which is currently 'In progress (0/5)'.

Amazon SageMaker

Getting started

Control panel

- Studio
- Studio Lab
- Canvas
- RStudio

SageMaker dashboard

- Images
- Lifecycle configurations
- Search

▼ JumpStart

- Foundation models

Automatic scaling was configured for variant AllTraffic

Amazon SageMaker > Endpoints > pytorch-inference-2022-11-28-18-38-11-439

pytorch-inference-2022-11-28-18-38-11-439

Delete

Endpoint settings

Name	pytorch-inference-2022-11-28-18-38-11-439	Type	Real-time
ARN	arn:aws:sagemaker:us-east-1:873635364805:endpoint/pytorch-inference-2022-11-28-18-38-11-439	Last updated	Mon Nov 28 2022 20:40:25 GMT+0200 (Eastern European Standard Time)
Status	InService	URL	https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2022-11-28-18-38-11-439/invocations
Creation time			Learn more about the API

Configuration

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Monitoring and operations tools

Concurrency

Concurrency

Edit

Function concurrency

Use unreserved account concurrency

Unreserved account concurrency

995

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](#)

Find configuration

Qualifier	Type	Provisioned concurrency	Status	Details
1	version	0	In progress (0/5)	-