

Operationalizing an AWS ML Project

Table of Contents

Notebook Setup	1
Sagemaker Training and Deployment	2
EC2 Training	3
Lambda functions	4
Security and Testing	5
Concurrency and Auto-scaling	6

Notebook Setup

I have chosen the 'ml.t2.medium' instance type for the following reasons. The execution of the code does not require a very computationally powerful CPU and high RAM, hence we should look at smaller instances. To avoid high costs, we should select a notebook that is low in per hour cost while meeting the require CPU and RAM needs. Comparing 'ml.t2.medium' and 'ml.t3.medium', the former is cheaper due to slower boot time while having the same 2 vCPU and 4GB memory. Since boot time speed is not important, the former was chosen.

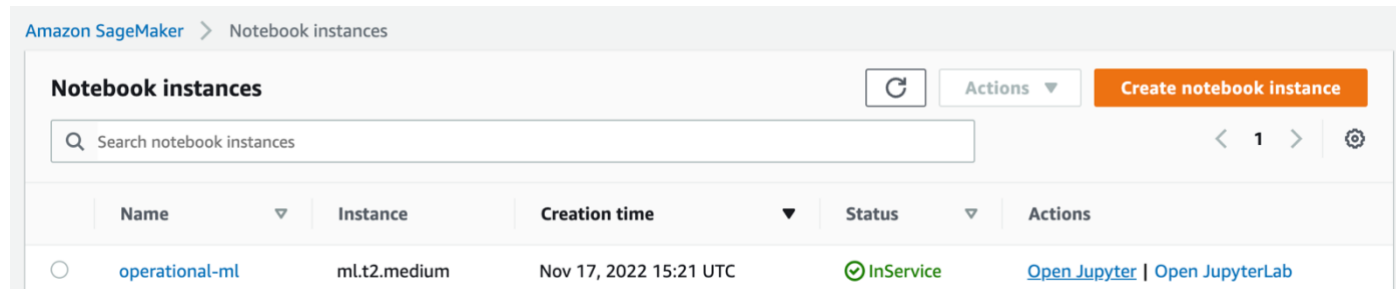


Figure 1. Sagemaker Notebook Instance

The dog breed dataset has been uploaded to the S3 bucket using sagemaker.

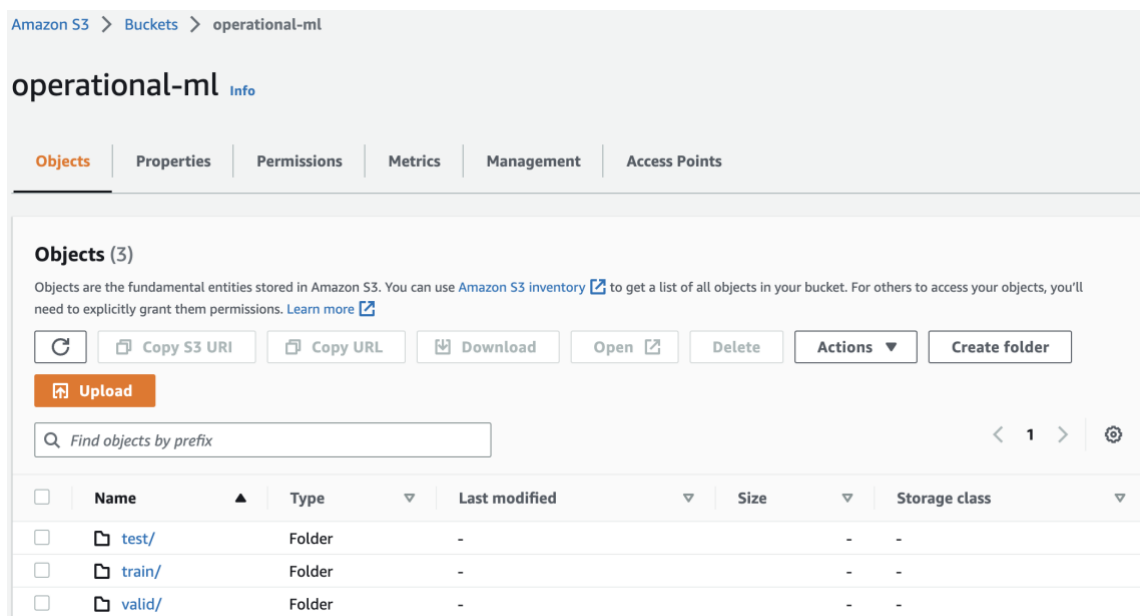


Figure 2. S3 Bucket

Sagemaker Training and Deployment

For hyperparameter tuning, the 'ml.m5.2xlarge' instance type, which has 8 vCPU and 32 GB of RAM at a cost of \$0.461 per hour, was used. Training was done using a multi-instance format, with max jobs of 6 and max parallel jobs of 3.

Hyperparameter tuning jobs					
<input type="text" value="Search hyperparameter tuning jobs"/>					
Name	Status	Training completed/total	Creation time	Duration	
pytorch-training-221117-1629	Completed	6 / 6	Nov 17, 2022 16:29 UTC	27 minutes	
pytorch-training-221117-1624	Completed	2 / 2	Nov 17, 2022 16:24 UTC	15 minutes	

Figure 3. Hyperparameter Tuning Job

For training, the 'ml.m5.2xlarge' instance type, which has 8 vCPU and 32 GB of RAM at a cost of \$0.461 per hour, was used. Training was done using a multi-instance format, with max jobs of 6 and max parallel jobs of 3.

Training jobs Info						
<input type="text" value="Search training jobs"/>				Refresh	Actions	Create training job
< 1 ... > Settings						
	Name	Creation time	Duration	Job status	Warm pool status	
<input type="radio"/>	dog-pytorch-2022-11-17-17-06-13-655	Nov 17, 2022 17:06 UTC	16 minutes	✔ Completed	-	
<input type="radio"/>	dog-pytorch-2022-11-17-17-06-11-307	Nov 17, 2022 17:06 UTC	15 minutes	✔ Completed	-	
<input type="radio"/>	pytorch-training-221117-1629-006-7b54c1eb	Nov 17, 2022 16:44 UTC	12 minutes	✔ Completed	⊖ Terminated	
<input type="radio"/>	pytorch-training-221117-1629-005-c0731b73	Nov 17, 2022 16:44 UTC	12 minutes	✔ Completed	⊖ Terminated	
<input type="radio"/>	pytorch-training-221117-1629-004-d56a386d	Nov 17, 2022 16:44 UTC	11 minutes	✔ Completed	⊖ Terminated	
<input type="radio"/>	pytorch-training-221117-1629-003-32b3c6d0	Nov 17, 2022 16:29 UTC	13 minutes	✔ Completed	⊖ Reused	
<input type="radio"/>	pytorch-training-221117-1629-002-be8064c1	Nov 17, 2022 16:29 UTC	14 minutes	✔ Completed	⊖ Reused	
<input type="radio"/>	pytorch-training-221117-1629-001-b52fa731	Nov 17, 2022 16:29 UTC	13 minutes	✔ Completed	⊖ Reused	

Figure 4. Multi-Instance Training Job

Multi instance deployed endpoint: [pytorch-inference-2022-11-17-17-22-57-546](#)

Endpoints					
<input type="text" value="Search endpoints"/>				Refresh	Update endpoint
Actions					
Create endpoint					
< 1 > Settings					
	Name	ARN	Creation time	Status	Last updated
<input type="radio"/>	pytorch-inference-2022-11-17-17-22-57-546	arn:aws:sagemaker:us-east-1:822767915126:endpoint/pytorch-inference-2022-11-17-17-22-57-546	Nov 17, 2022 17:22 UTC	✔ InService	Nov 17, 2022 17:25 UTC

Figure 5. Sagemaker Endpoints

EC2 Training

The **t2.xlarge** instance and the **Deep Learning AMI (Amazon Linux 2)** was used. Given that t2 instances can sustain high CPU performance for long periods without incurring extraordinary costs, this instance is a good mix of performance and affordability.

EC2 > Instances > i-0d16c00d2a86111f5		
Instance summary for i-0d16c00d2a86111f5 (operational-ml) Info		
Updated less than a minute ago		
Instance ID <input type="checkbox"/> i-0d16c00d2a86111f5 (operational-ml)	Public IPv4 address <input type="checkbox"/> 34.230.58.142 open address	Private IPv4 addresses <input type="checkbox"/> 172.0.1.67
IPv6 address -	Instance state ✔ Running	Public IPv4 DNS <input type="checkbox"/> ec2-34-230-58-142.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-0-1-67.ec2.internal	Private IP DNS name (IPv4 only) <input type="checkbox"/> ip-172-0-1-67.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.xlarge	

Figure 6. EC2 Instance Info

```
[root@ip-172-0-1-67 ~]# source activate pytorch
NOTE that the Amazon EC2 t2.xlarge instance type is not supported by current Deep Learning AMI. Please review the DLAMI release notes https://docs.aws.amazon.com/dlami/latest/devguide/appendix-ami-release-notes.html for supported Amazon EC2 instance types.
(pytorch) [root@ip-172-0-1-67 ~]# python solution.py
/opt/conda/envs/pytorch/lib/python3.9/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/python3.9/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, 115MB/s]
Starting Model Training
saved
(pytorch) [root@ip-172-0-1-67 ~]# ls
dogImages dogImages.zip solution.py TrainedModels
(pytorch) [root@ip-172-0-1-67 ~]# cd TrainedModels/
(pytorch) [root@ip-172-0-1-67 TrainedModels]# ls
model.pth
```

Figure 7. EC2 Training Saved - model.pth

Item	EC2	Sagemaker
Dataset/Model	Takes from local path	Takes from S3 bucket
Hyperparameter Tuning	Internal script	External script
Instances	Script and training job on the same instance	Script and training job on separate instance

Figure 8. Comparison of EC2 and Sagemaker scripts

Lambda functions

The lambda function will be used to invoke the deploy endpoint for the multi-instance trained model.

Lambda > Functions > ml-deploy

ml-deploy

Function overview

Info

ml-deploy

Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

23 seconds ago

Function ARN

arn:aws:lambda:us-east-1:822767915126:function:ml-deploy

Code source

Info

Upload from

File Edit Find View Go Tools Window

Test Deploy

Go to Anything (% P)

ml-deploy - /

_MACOSX

lambda_function.py

lambda_function

```

1
2 import base64
3 import logging
4 import json
5 import boto3
6 #import numpy
7 logger = logging.getLogger(__name__)
8 logger.setLevel(logging.DEBUG)
9
10 print('Loading Lambda function')
11
12 runtime=boto3.Session().client('sagemaker-runtime')
13 endpoint_Name='BradTestEndpoint'
14
15 def lambda_handler(event, context):
16
17     #x=event['content']
18     #aa=x.encode('ascii')
19     #bs=base64.b64decode(aa)
20     print('Context:::',context)
21     print('EventType:::',type(event))
22     bs=event
23     runtime=boto3.Session().client('sagemaker-runtime')

```

Figure 9. Lambda Function

Security and Testing

Test event was executed in lambda function.

Test event action

☐ Create new event

☒ Edit saved event

Event name

test_invoke

↻

Delete

Event JSON

Format JSON

```

1 {
2   "url": "https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2017/11/20/33-dog-breeds.jpg"
3 }
  
```

Figure 10. Lambda Function Test Event

Upon execution of test event, we got 'AccessDeniedException' as the lambda function does not have access to Sagemaker.

▼ Execution results

Status: Failed Max memory used: 69 MB Time: 779.00 ms

Test Event Name
 test_invoke

Response

```

{
  "errorMessage": "An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::822767915126:assumed-i",
  "errorType": "ClientError",
  "requestId": "e517f799-7eb4-4949-abd0-dffa675e4318",
  "stackTrace": [
    " File \"/var/task/lambda_function.py\", line 25, in lambda_handler\n      response=runtime.invoke_endpoint(EndpointName='pytorch-inference-2f",
    " 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"
  ]
}
  
```

Figure 11. Lambda function test event failure response

The 'SageMakerFullAccess' policy was added to lambda function's role.

Permissions policies (2) [Info](#)

↻

Simulate

Remove

Add permissions ▼

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	⊕ AWSLambdaBasicExecutionRole-851a92f8-ff6a-4a2a-a2f4-c96db...	Customer managed	
<input type="checkbox"/>	⊕ AmazonSageMakerFullAccess	AWS managed	Provides full access to Amazon SageMaker

Figure 12. Lambda function role IAM permissions

Test event was successfully executed, and 33 dog breeds was indicated in the result.

▼ Execution results Status: **Succeeded** Max memory used: 73 MB Time: 937.44 ms

Test Event Name
test_invoke

Response

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "Content-Type-In": "LambdaContext([aws_request_id=e5e9b125-1854-4ea6-8310-68388b572b61,log_group_name=/aws/lambda/ml-deploy,log_stream_name=20",
  "body": "[[-1.7426745891571045, -0.5980657339096069, -0.9946675300598145, -0.5695176124572754, -0.6992695331573486, -2.906853675842285, -0.332
```

Figure 13. Lambda function test event success response

While we have given the lambda function full access to sagemaker as well as the sagemaker notebook full access to S3, it is possible to add further granular permissions to allow access to a specific notebook or s3 bucket.

[IAM](#) > [Roles](#)

Roles (55) [Info](#) Refresh Delete Create role

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.

< 1 2 3 > ⚙️

<input type="checkbox"/>	Role name	Trusted entities	Last ac... ▼
<input type="checkbox"/>	AmazonSageMakerServiceCatalogProductsUseRole	AWS Service: cloudformation, and 9 more. ↗	19 minutes ago
<input type="checkbox"/>	AmazonSageMaker-ExecutionRole-20221031T235615	AWS Service: sagemaker	26 minutes ago
<input type="checkbox"/>	AmazonSageMaker-ExecutionRole-20220904T015471	AWS Service: sagemaker	45 minutes ago

Figure 14. IAM Roles

Concurrency and Auto-scaling

Version configuration for our lambda function was created.

Concurrency is set to 5, which means that the function can handle up to 5 requests at the same time. We will only be using provisioned concurrency, as we are not given the maximum number of requests.

Provisioned concurrency Refresh Edit Remove

Provisioned concurrency 0	Status 🔄 In progress (0/5)
------------------------------	-------------------------------

Figure 15. Lambda Function Provision Concurrency

Auto-scaling is set to a maximum of 3 instances, which means in time of high number of requests, 2 additional instances will be deployed. In addition, a scale-in and scale-out cooldown time of 30 seconds was used to ensure that users are not experiencing high latency.

Variant automatic scaling Learn more		
Variant name AllTraffic	Instance type ml.m5.large Elastic Inference -	Current instance count 1 Current weight 1
Minimum instance count	Maximum instance count	
1	3	

Figure 16. Endpoint Auto-scaling Config