Project Opeationalizing Machine Learning on SageMaker

Geraldo Margjini

June 15, 2025

Abstract

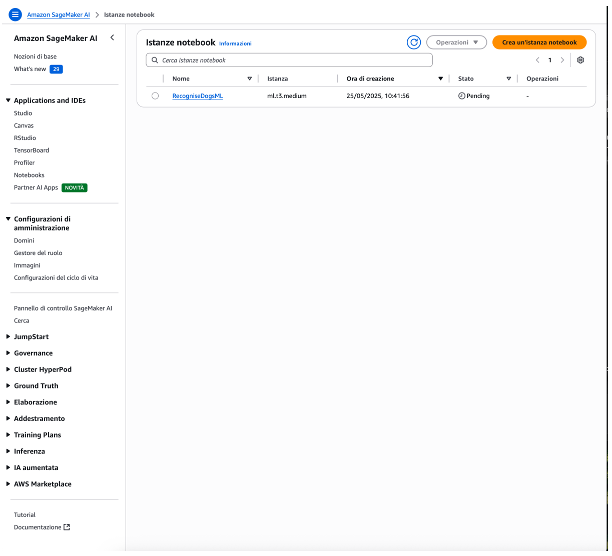
This document presents the write-up of the project, including answers and explanations to the assigned questions.

Contents

1 Step 1: Training and Deployment on Sagemaker

1.1 Initial Steps

I selected an ml.t3.medium instance because the dataset I'm working with is small. This instance type is among the most cost-effective options and is sufficient for the task at hand. Moreover, it falls within my AWS Free Tier usage.

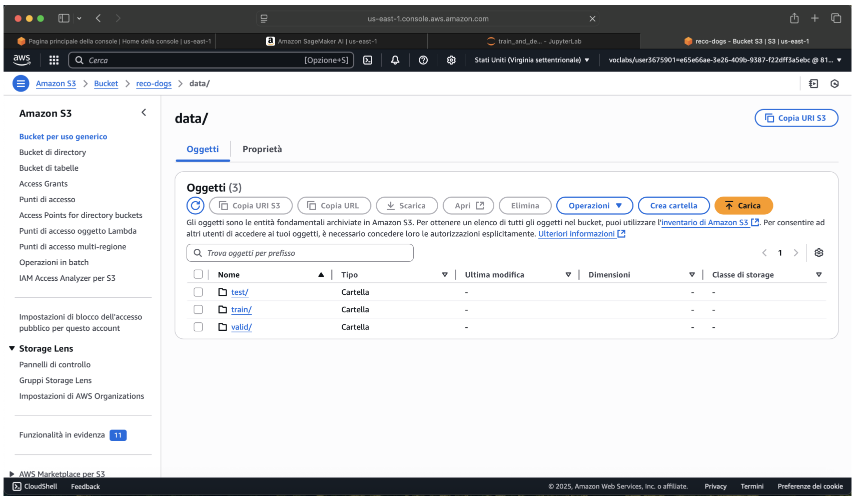


Nome	Istanza	Ora di creazione	Stato	Operazioni
RecogniseDogsML	ml.t3.medium	25/05/2025, 10:41:56	Pending	-

Figure 1: screenshot showing a created notebook instance on Sagemaker.

1.2 Download the data in the S3 Bucket

So the next step is to create a new S3 Bucket that i called *reco-dogs* and run the three first cell to download the data of different dog breads (133 dog breads, 6680 training, 835 validation and 836 test images).



us-east-1.console.aws.amazon.com
Pagina principale della console | Home della console | us-east-1
Amazon SageMaker AI | us-east-1
train_and_de... - JupyterLab
reco-dogs - Bucket S3 | S3 | us-east-1
Cerca [Opzione+S]
Stati Uniti (Virginia settentrionale)
voclabs/user3675901=e65e66ae-3e26-409b-9387-f22dff3a5ebc @ 81...
Amazon S3 > Bucket > reco-dogs > data/
Amazon S3
Bucket per uso generico
Bucket di directory
Bucket di tabelle
Access Grants
Punti di accesso
Access Points for directory buckets
Punti di accesso oggetto Lambda
Punti di accesso multi-regione
Operazioni in batch
IAM Access Analyzer per S3
Impostazioni di blocco dell'accesso pubblico per questo account
Storage Lens
Pannelli di controllo
Gruppi Storage Lens
Impostazioni di AWS Organizations
Funzionalità in evidenza 11
AWS Marketplace per S3
data/
Copia URI S3
Oggetti
Proprietà
Oggetti (3)
Copia URI S3
Copia URL
Scarica
Apri
Elimina
Operazioni
Crea cartella
Carica
Trova oggetti per prefisso
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. o affiliate.
Privacy
Termini
Preferenze dei cookie

Gli oggetti sono le entità fondamentali archiviate in Amazon S3. Per ottenere un elenco di tutti gli oggetti nel bucket, puoi utilizzare l'inventario di Amazon S3. Per consentire ad altri utenti di accedere ai tuoi oggetti, è necessario concedere loro le autorizzazioni esplicitamente. Ulteriori informazioni

Nome	Tipo	Ultima modifica	Dimensioni	Classe di storage
test/	Cartella	-	-	-
train/	Cartella	-	-	-
valid/	Cartella	-	-	-

Figure 2: screenshot showing a created and downloaded the Data in my S3 Bucket.

1.3 Single Instance Training: Training and Deployment

I created a tuning job with the following parameter and got

```
{'batch_size': 256, 'learning_rate': '0.002896533293199136'}
```

as best hyperparameters.

```
tuner = HyperparameterTuner(
    estimator,
    objective_metric_name,
    hyperparameter_ranges,
    metric_definitions,
    max_jobs=2,
    max_parallel_jobs=1,  # you once have one ml.g4dn.xlarge instance available
    objective_type=objective_type
)
```

So then i trained using this parameter with a ml.m5.xlarge:

```
estimator = PyTorch(
    entry_point='hpo.py',
    base_job_name='dog-pytorch',
    role=role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
    framework_version='1.4.0',
    py_version='py3',
    hyperparameters=hyperparameters,
    ## Debugger and Profiler parameters
    rules = rules,
    debugger_hook_config=hook_config,
    profiler_config=profiler_config,
)
```

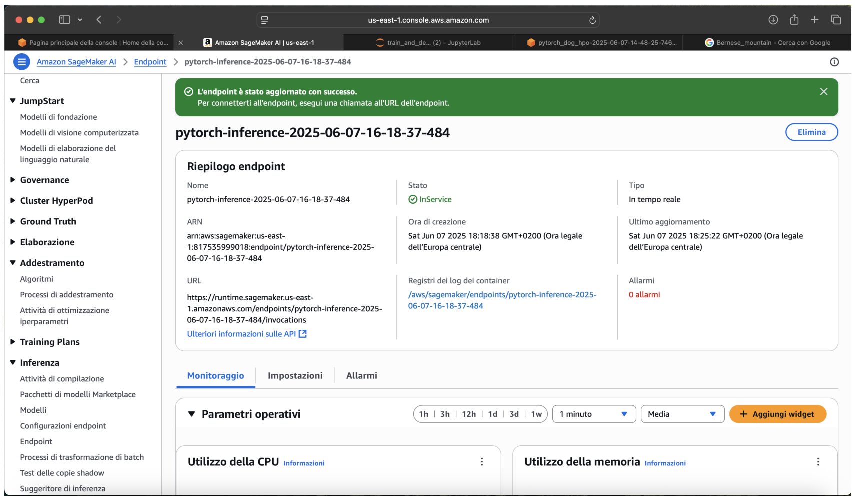
So following we have the screenshot of the Endpoint created:

1.4 Multy Instance Training: Training and Deployment

For the Multy instance i used the value for the

```
instance_count=4
```

to train four instances at time.


us-east-1.console.aws.amazon.com
Amazon SageMaker AI > Endpoint > pytorch-inference-2025-06-07-16-18-37-484
Cerca
JumpStart
Modelli di fondazione
Modelli di visione computerizzata
Modelli di elaborazione del linguaggio naturale
Governance
Cluster HyperPod
Ground Truth
Elaborazione
Addestramento
Algoritmi
Processi di addestramento
Attività di ottimizzazione iperparametri
Training Plans
Inferenza
Attività di compilazione
Pacchetti di modelli Marketplace
Modelli
Configurazioni endpoint
Endpoint
Processi di trasformazione di batch
Test delle copie shadow
Suggeritore di inferenza
Elimina
Monitoraggio
Impostazioni
Allarmi
Parametri operativi
1h 3h 12h 1d 3d 1w
1 minuto
Media
Aggiungi widget
Utilizzo della CPU Informazioni
Utilizzo della memoria Informazioni

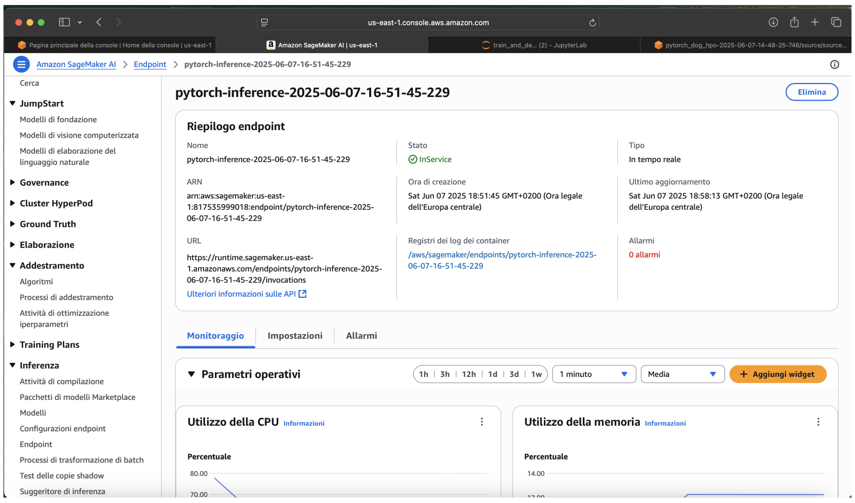
L'endpoint è stato aggiornato con successo.
Per connetterti all'endpoint, esegui una chiamata all'URL dell'endpoint.

pytorch-inference-2025-06-07-16-18-37-484

Riepilogo endpoint

Nome	Stato	Tipo
pytorch-inference-2025-06-07-16-18-37-484	InService	In tempo reale
ARN	**Ora di creazione**	**Ultimo aggiornamento**
arn:aws:sagemaker:us-east-1:817535999018:endpoint/pytorch-inference-2025-06-07-16-18-37-484	Sat Jun 07 2025 18:18:38 GMT+0200 (Ora legale dell'Europa centrale)	Sat Jun 07 2025 18:25:22 GMT+0200 (Ora legale dell'Europa centrale)
URL	**Registri dei log dei container**	**Allarmi**
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2025-06-07-16-18-37-484/invocations Ulteriori informazioni sulle API	/aws/sagemaker/endpoints/pytorch-inference-2025-06-07-16-18-37-484	0 allarmi

Figure 3: screenshot showing a created Endpoint for the single instance.


us-east-1.console.aws.amazon.com
Amazon SageMaker AI > Endpoint > pytorch-inference-2025-06-07-16-51-45-229
Cerca
JumpStart
Modelli di fondazione
Modelli di visione computerizzata
Modelli di elaborazione del linguaggio naturale
Governance
Cluster HyperPod
Ground Truth
Elaborazione
Addestramento
Algoritmi
Processi di addestramento
Attività di ottimizzazione iperparametri
Training Plans
Inferenza
Attività di compilazione
Pacchetti di modelli Marketplace
Modelli
Configurazioni endpoint
Endpoint
Processi di trasformazione di batch
Test delle copie shadow
Suggeritore di inferenza
Elimina
Monitoraggio
Impostazioni
Allarmi
Parametri operativi
1h 3h 12h 1d 3d 1w
1 minuto
Media
Aggiungi widget
Utilizzo della CPU Informazioni
Percentuale
80.00
70.00
Utilizzo della memoria Informazioni
Percentuale
14.00

pytorch-inference-2025-06-07-16-51-45-229

Riepilogo endpoint

Nome	Stato	Tipo
pytorch-inference-2025-06-07-16-51-45-229	InService	In tempo reale
ARN	**Ora di creazione**	**Ultimo aggiornamento**
arn:aws:sagemaker:us-east-1:817535999018:endpoint/pytorch-inference-2025-06-07-16-51-45-229	Sat Jun 07 2025 18:51:45 GMT+0200 (Ora legale dell'Europa centrale)	Sat Jun 07 2025 18:58:13 GMT+0200 (Ora legale dell'Europa centrale)
URL	**Registri dei log dei container**	**Allarmi**
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/pytorch-inference-2025-06-07-16-51-45-229/invocations Ulteriori informazioni sulle API	/aws/sagemaker/endpoints/pytorch-inference-2025-06-07-16-51-45-229	0 allarmi

Figure 4: screenshot showing a created Endpoint for the multy instance.

2 Step 2: EC2 Training

For this section i run an EC2 with training image:

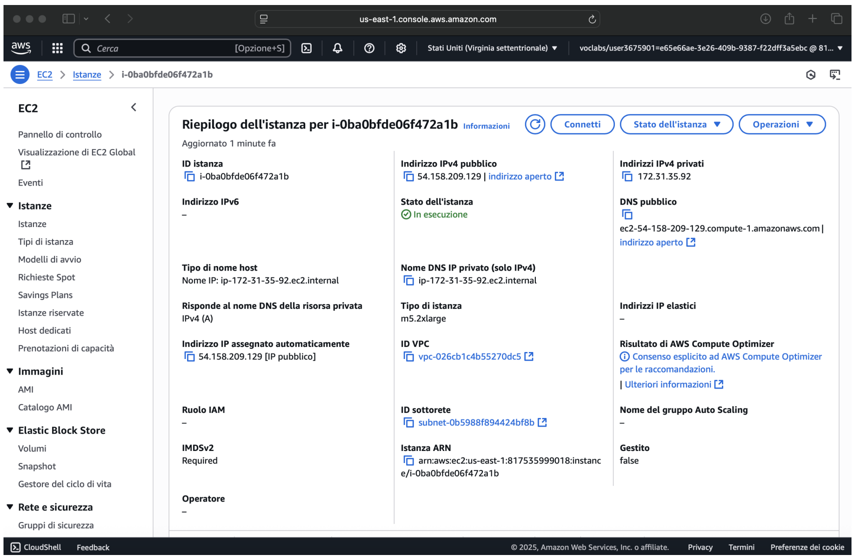
```
amazon/Deep Learning AMI Neuron (Amazon Linux 2023) 20250520
```

using an m5.2xlarge because it is a good tradeoff between the task that i have to accomplish and the costs.

So I started it with the WAS UI and connected it with an SSH:

```
ssh -i "ger.pem" ec2-user@ec2-54-158-209-129.compute-1.amazonaws.com
```

Once we are connected to our own EC2 we have to do the following step to: Download your own data, Create the target folder for the trained model, so we can copy the code for the train in a file called *solution.py* and run it.



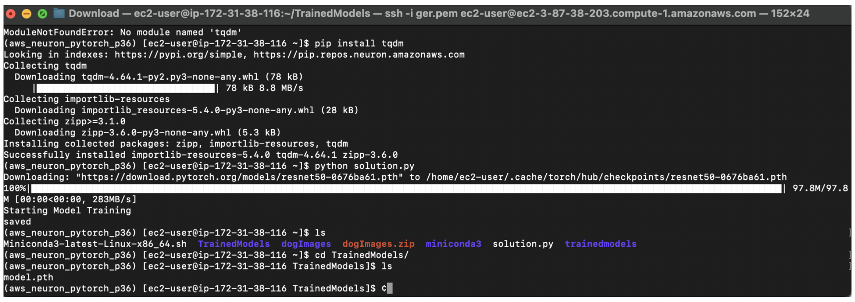
Riepilogo dell'istanza per i-0ba0bfde06f472a1b Informazioni

Aggiornato 1 minute fa

ID istanza i-0ba0bfde06f472a1b	**Indirizzo IPv4 pubblico** 54.158.209.129 | indirizzo aperto	**Indirizzi IPv4 privati** 172.31.35.92
Indirizzo IPv6 –	**Stato dell'istanza** In esecuzione	**DNS pubblico** ec2-54-158-209-129.compute-1.amazonaws.com | indirizzo aperto
Tipo di nome host Nome IP: ip-172-31-35-92.ec2.internal	**Nome DNS IP privato (solo IPv4)** ip-172-31-35-92.ec2.internal	
Risponde al nome DNS della risorsa privata IPv4 (A)	**Tipo di istanza** m5.2xlarge	**Indirizzi IP elastici** –
Indirizzo IP assegnato automaticamente 54.158.209.129 [IP pubblico]	**ID VPC** vpc-026cb1c4b55270dc5	**Risultato di AWS Compute Optimizer** Consenso esplicito ad AWS Compute Optimizer per le raccomandazioni. | Ulteriori informazioni
Ruolo IAM –	**ID sottorete** subnet-0b5988f894424bf8b	**Nome del gruppo Auto Scaling** –
IMDSv2 Required	**Istanza ARN** arn:aws:ec2:us-east-1:817535999018:instance/i-0ba0bfde06f472a1b	**Gestito** false
Operatore –		

Figure 5: screenshot showing a created EC2.

```
wget https://s3-us-west-1.amazonaws.com/udacity-aind/dog project/dogImages.zip
unzip dogImages.zip
mkdir TrainedModel
nano solution.py
python solution.py
```



```
Download — ec2-user@ip-172-31-38-116:~/TrainedModels — ssh -i ger.pem ec2-user@ec2-3-87-38-203.compute-1.amazonaws.com — 152×24
ModuleNotFoundError: No module named 'tqdm'
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 ~]$ pip install tqdm
Looking in indexes: https://pypi.org/simple, https://pip.repos.neuron.amazonaws.com
Collecting tqdm
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 8.8 MB/s
Collecting importlib-resources
  Downloading importlib_resources-5.4.0-py3-none-any.whl (28 kB)
Collecting zipp>=3.1.0
  Downloading zipp-3.6.0-py3-none-any.whl (5.3 kB)
Installing collected packages: zipp, importlib-resources, tqdm
Successfully installed importlib-resources-5.4.0 tqdm-4.64.1 zipp-3.6.0
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 ~]$ python solution.py
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to /home/ec2-user/.cache/torch/hub/checkpoints/resnet50-0676ba61.pth
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 97.8M/97.8
M [00:00<00:00, 283MB/s]
Starting Model Training
saved
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 ~]$ ls
Miniconda3-latest-Linux-x86_64.sh  TrainedModels  dogImages  dogImages.zip  miniconda3  solution.py  trainedmodels
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 ~]$ cd TrainedModels/
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 TrainedModels]$ ls
model.pth
(aws_neuron_pytorch_p36) [ec2-user@ip-172-31-38-116 TrainedModels]$ ¢
```

Figure 6: screenshot showing a finished training model.

I found some problem with the env that i had to create and install different missing pips.

2.1 Differances between the Sagemaker Notebook and EC2

In the Sagemaker we create a Nootebook instance where we write the code to create and instance to perform the training but in the EC2 we directly run the script code "localy" in the EC2.

- In the Sagemaker the hyperparameters are passed to hpo.py as an argument, but in the EC2 we run directly the hpo in the script.
- Using the EC2 we limited just to this instance but if we used the Sagemaker Notenook we can used the coded seen befored to perform distribuited trainig

- The final model, in the EC2 we have the model saved in then target folder, if we would like to save it in the could we should run the code wo upload it in the S3 bucket but using the Sagemaker Nootebook the "artifact" with is the model and the hpo are directly saved in a S3 target foolder.

3 Step 3: Lambda Fuction

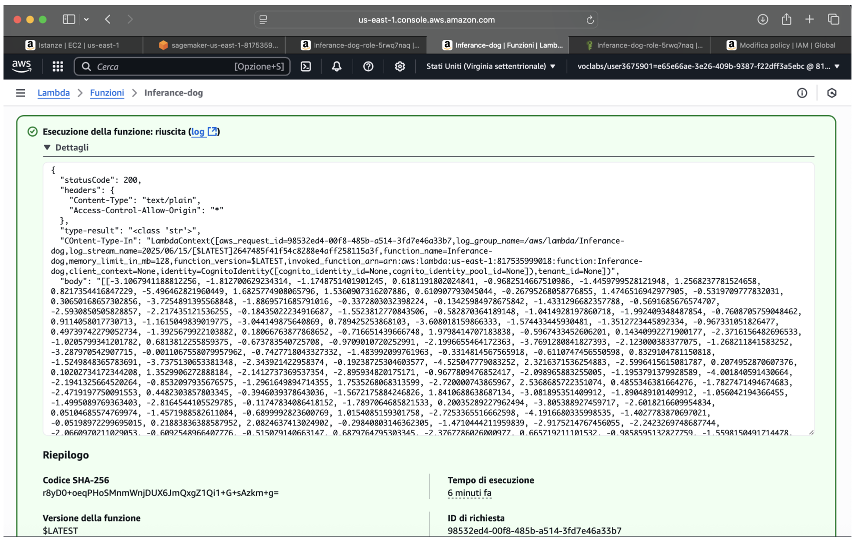
So the Lambda function created takes the endpoint created in the previous steps and makes inferences passing the S3 url of it. So in this i passed the name of my endpoint to my lambda function: *pytorch-inference-2025-06-15-15-57-32-476*.

4 Step 4: Lambda Security and Testing

Now i sended to the Endpoint this image path to do an inference:

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

Which is the same one used to do inference in the previous steps, for the security consideration i created a restricted policy to have just the specific Endpoint Access so this LambdaFuction can't perform other action to my other Endpoints.


us-east-1.console.aws.amazon.com
Istanze | EC2 | us-east-1
sagemaker-us-east-1-8175359...
Inferance-dog-role-5rwq7naq |...
Inferance-dog | Funzioni | Lamb...
Inferance-dog-role-5rwq7naq |...
Modifica policy | IAM | Global
Cerca
[Opzione+S]
Stati Uniti (Virginia settentrionale)
voclabs/user3675901=e65e66ae-3e26-409b-9387-f22dff3a5ebc @ 81...
Lambda
Funzioni
Inferance-dog

Esecuzione della funzione: riuscita (log)

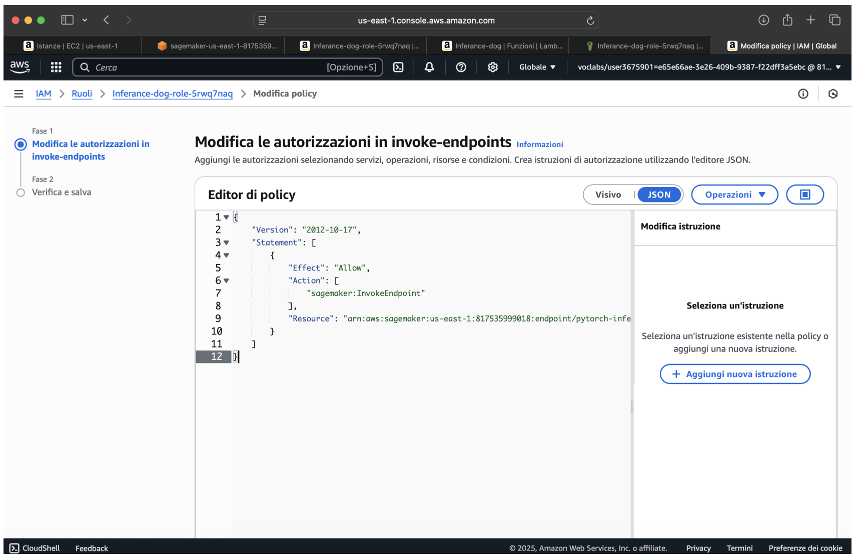
Dettagli

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "text/plain",
    "Access-Control-Allow-Origin": "*"
  },
  "type-result": "<class 'str'>",
  "COntent-Type-In": "LambdaContext([aws_request_id=98532ed4-00f8-485b-a514-3fd7e46a33b7,log_group_name=/aws/lambda/Inferance-
dog,log_stream_name=2025/06/15/[$LATEST]2647485f41f54c8288e4aff258115a3f,function_name=Inferance-
dog,memory_limit_in_mb=128,function_version=$LATEST,invoked_function_arn=arn:aws:lambda:us-east-1:817535999018:function:Inferance-
dog,client_context=None,identity=CognitoIdentity([cognito_identity_id=None,cognito_identity_pool_id=None]),tenant_id=None])",
  "body": "[[-3.1067941188812256, -1.812700629234314, -1.1748751401901245, 0.6181191802024841, -0.9682514667510986, -1.4459799528121948, 1.2568237781524658,
0.8217354416847229, -5.496462821960449, 1.6825774908065796, 1.5360907316207886, 0.610907793045044, -0.26795268058776855, 1.4746516942977905, -0.5319709777832031,
0.30650168657302856, -3.7254891395568848, -1.8869571685791016, -0.3372803032398224, -0.13425984978675842, -1.4331296682357788, -0.5691685676574707,
-2.593085050582857, -2.217435121536255, -0.18435022234916687, -1.5523812770843506, -0.582870364189148, -1.0414928197860718, -1.992409348487854, -0.7608705759048462,
0.9114058017730713, -1.1615049839019775, -3.044149875640869, 0.789425253868103, -3.608018159866333, -1.574433445930481, -1.3512723445892334, -0.967331051826477,
0.49739742279052734, -1.3925679922103882, 0.18066763877868652, -0.716651439666748, 1.9798414707183838, -0.5967433452606201, 0.14340992271900177, -2.3716156482696533,
-1.0205799341201782, 0.6813812255859375, -0.673783540725708, -0.9709010720252991, -2.1996655464172363, -3.7691280841827393, -2.123000383377075, -1.268211841583252,
-3.287970542907715, -0.0011067558079957962, -0.7427718043327332, -1.483992099761963, -0.3314814567565918, -0.6110747456550598, 0.8329104781150818,
-1.5249848365783691, -3.7375130653381348, -2.343921422958374, -0.19238725304603577, -4.525047779083252, 2.3216371536254883, -2.5996415615081787, 0.2074952870607376,
0.10202734172344208, 1.3529906272888184, -2.1412737369537354, -2.895934820175171, -0.9677809476852417, -2.098965883255005, -1.1953791379928589, -4.001840591430664,
-2.1941325664520264, -0.8532097935676575, -1.2961649894714355, 1.7535268068313599, -2.720000743865967, 2.5368685722351074, 0.4855346381664276, -1.7827471494674683,
-2.4719197750091553, 0.4482303857803345, -0.3946039378643036, -1.5672175884246826, 1.8410688638687134, -3.081895351409912, -1.890489101409912, -1.056042194366455,
-1.4995089769363403, -2.816454410552978‍5, -0.11747834086418152, -1.7897064685821533, 0.20035289227962494, -3.805388927459717, -2.6018216609954834,
0.05104685574769974, -1.4571988582611084, -0.6899992823600769, 1.0154085159301758, -2.7253365516662598, -4.1916680335998535, -1.4027783870697021,
-0.05198972299695015, 0.21883836388587952, 2.0824637413024902, -0.29840803146362305, -1.4710444211959839, -2.9175214767456055, -2.2423269748687744,
-2.0660970211029053, -0.6092548966407776, -0.515079140663147, 0.6879764795303345, -2.3767786026000977, 0.665719211101532, -0.9858595132827759, -1.5598150491714478,
```

Riepilogo

Codice SHA-256 r8yD0+oeqPHoSMnmWnjDUX6JmQxgZ1Qi1+G+sAzkm+g=	**Tempo di esecuzione** 6 minuti fa
Versione della funzione $LATEST	**ID di richiesta** 98532ed4-00f8-485b-a514-3fd7e46a33b7

Figure 7: screenshot showing a inference made with my LambdaFuction.


Modifica le autorizzazioni in invoke-endpoints

Aggiungi le autorizzazioni selezionando servizi, operazioni, risorse e condizioni. Crea istruzioni di autorizzazione utilizzando l'editore JSON.

Editor di policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
            "Resource": "arn:aws:sagemaker:us-east-1:817535999018:endpoint/pytorch-infe
        }
    ]
}
```

Seleziona un'istruzione

Seleziona un'istruzione esistente nella policy o aggiungi una nuova istruzione.

us-east-1.console.aws.amazon.com
IAM > Ruoli > Inferance-dog-role-5rwq7naq > Modifica policy
Fase 1
Modifica le autorizzazioni in invoke-endpoints
Fase 2
Verifica e salva
Visivo
JSON
Operazioni
Modifica istruzione
Aggiungi nuova istruzione

Figure 8: screenshot showing the specific policy for my LambdaFunction.

5 Step 5: Concurrency and Auto-Scaling

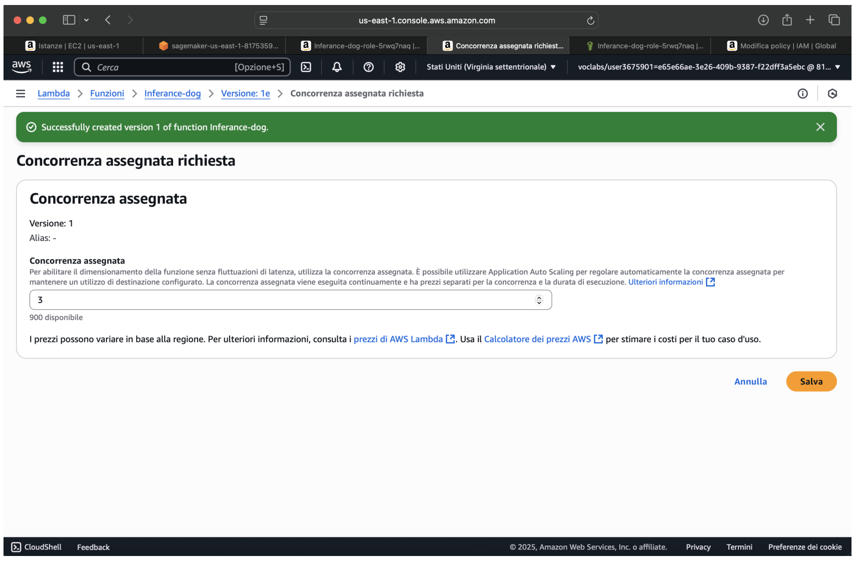
5.1 Concurrency

Concurrency refers to the ability of Lambda functions to handle multiple requests simultaneously.

We can configure either Reserved or Provisioned concurrency for our function. Provisioned concurrency provides faster response times but comes with higher cost.

Because our expected request volume is low, we don't need very high concurrency. I have set Provisioned concurrency to 3, which is sufficient for our needs, and Reserved concurrency to 100.

Below is a screenshot of the Lambda concurrency settings:


Successfully created version 1 of function Inferance-dog.

Concorrenza assegnata richiesta

Concorrenza assegnata

Versione: 1

Alias: -

Concorrenza assegnata

Per abilitare il dimensionamento della funzione senza fluttuazioni di latenza, utilizza la concorrenza assegnata. È possibile utilizzare Application Auto Scaling per regolare automaticamente la concorrenza assegnata per mantenere un utilizzo di destinazione configurato. La concorrenza assegnata viene eseguita continuamente e ha prezzi separati per la concorrenza e la durata di esecuzione. Ulteriori informazioni

3

900 disponibile

I prezzi possono variare in base alla regione. Per ulteriori informazioni, consulta i prezzi di AWS Lambda. Usa il Calcolatore dei prezzi AWS per stimare i costi per il tuo caso d'uso.

us-east-1.console.aws.amazon.com
Lambda > Funzioni > Inferance-dog > Versione: 1e > Concorrenza assegnata richiesta
Annulla
Salva

Figure 9: screenshot showing the concurrency settins.

5.2 Auto-scaling

Auto-scaling refers to the ability of endpoints to service multiple lambda function requests at once. I chose to autoscale endpoints to 4 instances maximum, with scale in coold down time of 30 seconds and scale out cool down time of 300 seconds. These settings are sufficient for our project needs and workload.