5_SageMaker_Inference_Endpoint_Creation

December 12, 2025

0.1 5. SageMaker inference endpoint

Now that we have our trained model on the graph data we can deploy it as an inference endpoint to do predictions in real-time

We can utilize the resources provided by GraphStorm in the GraphStorm GitHub. They provide an image pushable to Amazon ECR containing the endpoint and all the dependencies needed to run the model.

[1]:

```
!bash ./graphstorm/docker/build_graphstorm_image.sh --environment sagemaker-endpoint --device cpu > /dev/null 2>&1
```

[2]:

```
!bash ./graphstorm/docker/push_graphstorm_image.sh --environment sagemaker-endpoint --device cpu --region eu-west-1 --account 992382462371
```

```
Execution parameters:
- ENVIRONMENT: sagemaker-endpoint
- DEVICE TYPE: cpu
- IMAGE: graphstorm
- TAG: sagemaker-endpoint-cpu
- REGION: eu-west-1
- ACCOUNT: 992382462371
Getting or creating container repository: graphstorm
WARNING: ECR repository graphstorm does not exist in region eu-west-1.
Attempting to create…
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:eu-
west-1:992382462371:repository/graphstorm",
        "registryId": "992382462371",
        "repositoryName": "graphstorm",
        "repositoryUri": "992382462371.dkr.ecr.eu-
west-1.amazonaws.com/graphstorm",
        "createdAt": 1764514964.791,
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
Successfully created ECR repository graphstorm
Logging into ECR with local credentials
WARNING! Your password will be stored unencrypted in
/home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
Pushing image to 992382462371.dkr.ecr.eu-
west-1.amazonaws.com/graphstorm:sagemaker-endpoint-cpu
The push refers to repository [992382462371.dkr.ecr.eu-
west-1.amazonaws.com/graphstorm]

fc83a2be: Preparing
6eab277e: Preparing
b6c0760e: Preparing
4d6805b7: Preparing
2bf9ae09: Preparing
cbbe6821: Preparing
1b100228: Preparing
1b1aeb17: Preparing
9269f288: Preparing
2502d680: Preparing
1f29f59d: Preparing
b0a525a2: Preparing
22369f83: Preparing
f5eb30fd: Preparing
b2fa1d94: Preparing
e70272ac: Preparing
ce92b8d8: Preparing
269f288: Waiting g
b1aeb17: Waiting g
4ab58b6c: Preparing
502d680: Waiting g
0a525a2: Waiting g
2369f83: Waiting g
2fa1d94: Waiting g
70272ac: Waiting g
bbe6821: Waiting g
cfb8d2b: Waiting g
4ab58b6c: Waiting g
1d31ad99: Preparing
6a033de4: Waiting g
ebc7b88d: Waiting g
4d43e8ba: Preparing
```

```
d01c10db: Waiting g
4d43e8ba: Waiting g
517abf3d: Waiting g
5584aa37: Waiting g
5150b5bb: Preparing
670fdd55: Waiting g
5150b5bb: Waiting g
sagemaker-endpoint-cpu: digest:
sha256:82e3f414ed7e789c07ad3a59c4dccd6a5ad9a4e6ab7f1d0342aa523a95d88411 size:
8698
```

[11]:

```
# You need an S3 location to upload the model artifacts to
S3_BUCKET = 'tfm-fraud-detection-anna-model-dub'
# The endpoint needs an execution role with a number of permissions that allow␣
 ↪it to function
ENDPOINT_ROLE = 'arn:aws:iam::992382462371:role/fraud-detection'
# We are deploying the endpoint within the same VPC as the NeptuneDB cluster,␣
 ↪so we need that information at deployment time
VPC_SUBNET_IDS = 'subnet-09065d808acfd58a4'
VPC_SECURITY_GROUP_IDS = 'sg-072d6e4422eb74799'

ACCOUNT_ID = '992382462371'
AWS_REGION = 'eu-west-1'
MODEL_PATH = './model-simple-hgt'

# Build up training command from variables
command = f"""python graphstorm/sagemaker/launch/launch_realtime_endpoint.py \
        --image-uri "{ACCOUNT_ID}.dkr.ecr.{AWS_REGION}.amazonaws.com/graphstorm:
 ↪sagemaker-endpoint-cpu" \
        --role {ENDPOINT_ROLE} \
        --region {AWS_REGION} \
        --instance-type ml.c6i.xlarge \
        --restore-model-path {MODEL_PATH}/epoch-1 \
        --model-yaml-config-file {MODEL_PATH}/
 ↪GRAPHSTORM_RUNTIME_UPDATED_TRAINING_CONFIG.yaml \
        --graph-json-config-file {MODEL_PATH}/data_transform_new.json \
        --infer-task-type node_classification \
        --upload-tarfile-s3 s3://{S3_BUCKET}/model-artifacts \
        --model-name ieee-fraud-detection \
        --vpc-subnet-ids {VPC_SUBNET_IDS} \
        --vpc-security-group-ids {VPC_SECURITY_GROUP_IDS} \
        --async-execution false"""
```

[12]:

```
!{command}
```

```
sagemaker.config INFO - Not applying SDK defaults from location:
/etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location:
```

```
/home/ec2-user/.config/sagemaker/config.yaml
INFO:botocore.credentials:Found credentials from IAM Role:
BaseNotebookInstanceEc2InstanceRole
INFO:botocore.credentials:Found credentials from IAM Role:
BaseNotebookInstanceEc2InstanceRole
Waiting for endpoint 'ieee-fraud-detection-Endpoint-2025-11-30-15-22-04' to be
in service in eu-west-1 region…
Endpoint named 'ieee-fraud-detection-Endpoint-2025-11-30-15-22-04' has been
successfully created, and ready to be invoked!
```

The above script with create an endpoint from the Docker created previously and the model artifacts

[1]:

```python
import json
import time
import yaml

import boto3
import requests
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from requests.adapters import HTTPAdapter
from urllib3.util import Retry
import urllib3
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest

# Configure retry strategy
retry_strategy = Retry(total=3, backoff_factor=1, status_forcelist=[500, 502,␣
 ↪503, 504])

# Set up AWS credentials for request signing
session = boto3.Session()
credentials = session.get_credentials()
region = session.region_name

# Set up session with retry
urllib3.disable_warnings()
http_session = requests.Session()
adapter = HTTPAdapter(max_retries=retry_strategy)
http_session.mount("https://", adapter)
http_session.verify = False

# Neptune endpoint configuration
NEPTUNE_HOST = 'tfm-fraud-detection.cluster-crqqiey689tq.eu-west-1.neptune.
 ↪amazonaws.com'
NEPTUNE_READER_ENDPOINT = 'tfm-fraud-detection.cluster-ro-crqqiey689t.eu-west-1.
 ↪neptune.amazonaws.com'
```

```
GREMLIN_PORT = 8182
```

[ ]:

```
```