3_Loading_to_Neptune

December 12, 2025

0.1 3. Loading Data into Neptune

[4]:
```
import logging
import os
import json
import time

import boto3
import requests
import urllib3
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
from requests.adapters import HTTPAdapter
from botocore.session import Session
from requests.auth import AuthBase
from urllib3.util import Retry
```

0.1.1 3.1 Uploading data to S3

We will first upload the data to an S3 bucket accessible by the Neptune instance.

[7]:
```
S3_GRAPH_PREFIX = f"s3://tfm-fraud-detection-anna-2/neptune-input/
 ↪ieee-cis-with-text-embeddings"
NEPTUNE_ENDPOINT = 'tfm-fraud-detection.cluster-crqqiey689tq.eu-west-1.neptune.
 ↪amazonaws.com'
NEPTUNE_READER_ENDPOINT = 'tfm-fraud-detection.cluster-ro-crqqiey689tq.
 ↪eu-west-1.neptune.amazonaws.com'
NEPTUNE_PORT = 8182
NEPTUNE_HOST = 'tfm-fraud-detection.cluster-crqqiey689tq.eu-west-1.neptune.
 ↪amazonaws.com:8182'
```

[23]:
```
!aws s3 sync ./graph_data/ {S3_GRAPH_PREFIX}/edges/ --exclude "*" --include␣
 ↪"Edge*.csv"
!aws s3 sync ./graph_data/ {S3_GRAPH_PREFIX}/vertices/ --exclude "*" --include␣
 ↪"Vertex*.csv"
```

```
/bin/bash: switchml: line 1: syntax error: unexpected end of file
/bin/bash: error importing function definition for `switchml'
/bin/bash: module: line 1: syntax error: unexpected end of file
```

```
/bin/bash: error importing function definition for `module'
upload: graph_data/Edge_Transaction,identified_by,Address1_0.csv to s3://tfm-
fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Address1_0.csv
upload: graph_data/Edge_Transaction,identified_by,Address2_0.csv to s3://tfm-
fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Address2_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card3_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card3_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card2_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card2_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card1_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card1_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card4_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card4_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card5_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card5_0.csv
upload: graph_data/Edge_Transaction,identified_by,Card6_0.csv to s3://tfm-fraud-
detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,Card6_0.csv
upload: graph_data/Edge_Transaction,identified_by,ProductCD_0.csv to s3://tfm-
fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,ProductCD_0.csv
upload: graph_data/Edge_Transaction,identified_by,P_emaildomain_0.csv to
s3://tfm-fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,P_emaildomain_0.csv
upload: graph_data/Edge_Transaction,identified_by,R_emaildomain_0.csv to
s3://tfm-fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/Edge_Transaction,identified_by,R_emaildomain_0.csv
/bin/bash: switchml: line 1: syntax error: unexpected end of file
/bin/bash: error importing function definition for `switchml'
/bin/bash: module: line 1: syntax error: unexpected end of file
/bin/bash: error importing function definition for `module'
upload: graph_data/Vertex_Address2_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_Address2_0.csv
upload: graph_data/Vertex_Card4_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card4_0.csv
upload: graph_data/Vertex_Card3_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card3_0.csv
upload: graph_data/Vertex_ProductCD_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_ProductCD_0.csv
```

```
upload: graph_data/Vertex_Card5_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card5_0.csv
upload: graph_data/Vertex_Card6_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card6_0.csv
upload: graph_data/Vertex_Card2_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card2_0.csv
upload: graph_data/Vertex_R_emaildomain_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_R_emaildomain_0.csv
upload: graph_data/Vertex_P_emaildomain_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_P_emaildomain_0.csv
upload: graph_data/Vertex_Address1_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_Address1_0.csv
upload: graph_data/Vertex_Transaction_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/Vertex_Transaction_0.csv
upload: graph_data/Vertex_Card1_0.csv to s3://tfm-fraud-detection-
anna-2/neptune-input/ieee-cis-with-text-embeddings/vertices/Vertex_Card1_0.csv
```

0.1.2 3.2 Importing Data into Neptune

Now we will use the Neptune Loader API to import the data

[3]:
```
urllib3.disable_warnings()

session = requests.Session()
adapter = HTTPAdapter(max_retries=3)
session.mount("https://", adapter)
session.verify = False

boto3_session = boto3.Session()
credentials = boto3_session.get_credentials()
region = boto3_session.region_name
```

[4]:
```
loader_endpoint = f"https://{NEPTUNE_ENDPOINT}:{NEPTUNE_PORT}/loader"
```

[5]:
```
payload_vertices = {
    "source": f"{S3_GRAPH_PREFIX}/vertices/",
    "format": 'csv',
    "iamRoleArn": 'arn:aws:iam::992382462371:role/NeptuneLoadFromS3',
    "region": 'eu-west-1',
    "failOnError": "TRUE",
    "parallelism": "OVERSUBSCRIBE",
    "queueRequest": "TRUE",
    "updateSingleCardinalityProperties": "FALSE",
    "edgeOnlyLoad": "FALSE",
```

```
}
```

[6]:
```
payload_vertices
```

[6]:
```
{'source': 's3://tfm-fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/vertices/',
 'format': 'csv',
 'iamRoleArn': 'arn:aws:iam::992382462371:role/NeptuneLoadFromS3',
 'region': 'eu-west-1',
 'failOnError': 'TRUE',
 'parallelism': 'OVERSUBSCRIBE',
 'queueRequest': 'TRUE',
 'updateSingleCardinalityProperties': 'FALSE',
 'edgeOnlyLoad': 'FALSE'}
```

[7]:
```
try:
    request = AWSRequest(
        method="POST",
        url=loader_endpoint,
        data=json.dumps(payload_vertices),
        headers={"Content-Type": "application/json", "Host": NEPTUNE_HOST},
    )
    SigV4Auth(credentials, "neptune-db", region).add_auth(request)
    headers = dict(request.headers)

    response = session.post(
        loader_endpoint, headers=headers, json=payload_vertices, timeout=120
    )

    if response.status_code != 200:
        raise Exception(f"Failed to start load job: {response.text}")

    load_id = response.json()["payload"]["loadId"]
    print(f"Started load job with ID: {load_id}")
except requests.exceptions.RequestException as e:
    print(f"Error connecting to Neptune: {str(e)}")
    raise
```

```
Started load job with ID: 39cab024-ffa9-4525-baa9-676e4184d2da
```

[28]:
```
load_id = '39cab024-ffa9-4525-baa9-676e4184d2da'

region = "eu-west-1"
service = "neptune-db"
session = boto3.Session()
credentials = session.get_credentials().get_frozen_credentials()
```

```
loader_status_url = f"https://{NEPTUNE_ENDPOINT}:{NEPTUNE_PORT}/loader/
 ↪{load_id}"

request = AWSRequest(method="GET", url=loader_status_url, headers={})
SigV4Auth(credentials, service, region).add_auth(request)

signed_headers = dict(request.headers)

response = requests.get(loader_status_url, headers=signed_headers, timeout=30)

if response.status_code != 200:
    raise Exception(f"Failed to get load status: {response.text}")

status_response = response.json()
print(status_response)
```

```
{'status': '200 OK', 'payload': {'feedCount': [{'LOAD_COMPLETED': 12}],
'overallStatus': {'fullUri': 's3://tfm-fraud-detection-anna-2/neptune-
input/ieee-cis-with-text-embeddings/vertices/', 'runNumber': 1, 'retryNumber':
0, 'status': 'LOAD_COMPLETED', 'totalTimeSpent': 201, 'startTime': 1764514103,
'totalRecords': 13320891, 'totalDuplicates': 0, 'parsingErrors': 0,
'datatypeMismatchErrors': 0, 'insertErrors': 0}}}
```

[29]:
```
payload_edges = {
    "source": f"{S3_GRAPH_PREFIX}/edges/",
    "format": 'csv',
    "iamRoleArn": 'arn:aws:iam::992382462371:role/NeptuneLoadFromS3',
    "region": 'eu-west-1',
    "failOnError": "TRUE",
    "parallelism": "OVERSUBSCRIBE",
    "queueRequest": "TRUE",
    "updateSingleCardinalityProperties": "FALSE",
    "edgeOnlyLoad": "TRUE",
}
```

[30]:
```
payload_edges
```

[30]:
```
{'source': 's3://tfm-fraud-detection-anna-2/neptune-input/ieee-cis-with-text-
embeddings/edges/',
 'format': 'csv',
 'iamRoleArn': 'arn:aws:iam::992382462371:role/NeptuneLoadFromS3',
 'region': 'eu-west-1',
 'failOnError': 'TRUE',
 'parallelism': 'OVERSUBSCRIBE',
 'queueRequest': 'TRUE',
 'updateSingleCardinalityProperties': 'FALSE',
 'edgeOnlyLoad': 'TRUE'}
```

[33]:
```
session = requests.Session()
adapter = HTTPAdapter(max_retries=3)
session.mount("https://", adapter)
session.verify = False

try:
    request_e = AWSRequest(
        method="POST",
        url=loader_endpoint,
        data=json.dumps(payload_edges),
        headers={"Content-Type": "application/json", "Host": NEPTUNE_HOST},
    )
    SigV4Auth(credentials, "neptune-db", region).add_auth(request_e)
    headers = dict(request_e.headers)

    response = session.post(
        loader_endpoint, headers=headers, json=payload_edges, timeout=120
    )

    if response.status_code != 200:
        raise Exception(f"Failed to start load job: {response.text}")

    load_id = response.json()["payload"]["loadId"]
    print(f"Started load job with ID: {load_id}")
except requests.exceptions.RequestException as e:
    print(f"Error connecting to Neptune: {str(e)}")
    raise
```

```
Started load job with ID: 10ef6c9a-81d3-4f22-a69f-05556af32bbb
```

[40]:
```
# We monitor the job until it's completed

region = "eu-west-1"
service = "neptune-db"
load_id = '10ef6c9a-81d3-4f22-a69f-05556af32bbb'
session = boto3.Session()
credentials = session.get_credentials().get_frozen_credentials()

loader_status_url = f"https://{NEPTUNE_ENDPOINT}:{NEPTUNE_PORT}/loader/
 ↪{load_id}"

request = AWSRequest(method="GET", url=loader_status_url, headers={})
SigV4Auth(credentials, service, region).add_auth(request)

signed_headers = dict(request.headers)

response = requests.get(loader_status_url, headers=signed_headers, timeout=30)
```

```
if response.status_code != 200:
    raise Exception(f"Failed to get load status: {response.text}")

status_response = response.json()
print(status_response)
```

```
{'status': '200 OK', 'payload': {'feedCount': [{'LOAD_COMPLETED': 11}],
'overallStatus': {'fullUri': 's3://tfm-fraud-detection-anna-2/neptune-
input/ieee-cis-with-text-embeddings/edges/', 'runNumber': 1, 'retryNumber': 0,
'status': 'LOAD_COMPLETED', 'totalTimeSpent': 223, 'startTime': 1764514361,
'totalRecords': 6495940, 'totalDuplicates': 0, 'parsingErrors': 0,
'datatypeMismatchErrors': 0, 'insertErrors': 0}}}
```

[9]:
```
import requests
import json

# Neptune HTTPS endpoint (inside VPC or via SSH tunnel)
NEPTUNE_ENDPOINT_GREMLIN = f"https://{NEPTUNE_READER_ENDPOINT}:8182/gremlin"

session = boto3.Session()
credentials = session.get_credentials().get_frozen_credentials()

# Example Gremlin query: fetch one Address1 node
gremlin_query = {
    "gremlin": "g.V().hasLabel('Address1').limit(1).valueMap(true)"
}

request = AWSRequest(
    method="POST",
    url=NEPTUNE_ENDPOINT_GREMLIN,
    data=json.dumps(gremlin_query),
    headers={"Content-Type": "application/json"}
)

SigV4Auth(credentials, "neptune-db", "eu-west-1").add_auth(request)

prepared = request.prepare()
response = requests.post(prepared.url, headers=dict(prepared.headers),␣
 ↪data=prepared.body)


# Parse and print result
print(response.status_code)
print(response.json())
```

```
200
{'requestId': '9d0c9f5b-79e2-457b-9577-f77ee7d0417c', 'status': {'message': '',
```

'code': 200, 'attributes': {'@type': 'g:Map', '@value': []}}, 'result': {'data':
{'@type': 'g:List', '@value': [{'@type': 'g:Map', '@value': ['id_embeddings',
{'@type': 'g:List', '@value': ['-0.10411066|-0.09495294|-0.02560376|-
0.10471054|-0.03221983|-0.06918737|0.00671981|-0.02563450|-0.03906793|-
0.04467977|0.06368162|-0.11168611|-0.04701068|-0.04343639|-
0.04448557|0.03385739|-0.03590583|-0.08209366|0.01377341|-0.06490656|-
0.08229642|0.06850249|-0.03212234|0.00875039|0.03205270|-0.05482785|-
0.10645267|0.10052497|0.02506749|-0.11014227|0.07382924|-
0.01859366|0.09385327|0.05079322|0.08397678|-0.01902092|0.02725577|0.03003289|-
0.08220153|0.04295432|0.02004967|-0.01912498|0.01021252|0.04135024|-0.07979953|-
0.00384993|-0.01007226|-0.00608236|0.08176813|0.07039153|0.05313323|0.06642231|-
0.02136299|-0.01280364|0.03230818|0.00410934|-0.02523403|0.13457833|-
0.05438569|0.02600227|0.08023125|0.01405102|0.08256846|0.00219734|-0.03765880|-
0.02416829|-0.03135327|-0.03533762|-0.04390898|-0.04836375|-
0.01445918|0.05396505|-0.05052062|-0.02410694|-0.03451125|-0.02119962|-
0.01571608|-0.03221149|0.03457045|-0.00562834|-0.01436860|-
0.01958647|0.00465265|0.04145118|0.01337212|-0.01312762|0.01894267|-
0.01466292|0.07373560|0.01636299|-0.02890290|-0.02591410|-0.00145998|-
0.00061629|-0.00543539|0.03817476|0.08437544|-0.01285802|-
0.09707833|0.05693980|-0.02795737|-0.03031687|-0.02247824|0.07204594|-
0.02992309|-0.02471920|-0.00822410|0.04494400|-0.01104013|-
0.01200134|0.04450207|-0.03162053|0.00880048|-0.06109320|0.01172755|0.00202610|-
0.03320459|0.04036769|0.08311047|-0.08434557|0.04045083|-0.07228520|-
0.04646523|-0.08202411|-0.08850009|-0.04255259|0.00240611|-0.00000000|-
0.01549085|0.03031270|-0.09263222|0.10636815|0.04723730|0.01184574|-
0.01275808|0.08788544|-0.11738869|-0.05883644|-0.01529018|-0.04577078|-
0.01200626|0.01437020|-0.00283592|-0.03434550|0.04569695|0.02381887|-
0.04161311|-0.01024679|-0.01316378|0.01967588|-
0.05683175|0.02339353|0.02476498|0.12427209|-0.01475545|-0.06659744|0.08529634|-
0.01682968|0.05175595|-0.04617647|-
0.09282923|0.01866287|0.04349222|0.01054278|0.03479411|0.00224163|-0.01634230|-
0.04126639|-0.02493958|0.03982719|-0.08462235|-0.09682225|0.06447475|-
0.03200143|0.03065779|0.01110678|0.03874030|0.01197738|-0.06751479|-0.02860214|-
0.04354279|0.01880577|-0.10314863|-0.04328854|-0.04991994|-0.00005630|-
0.05142452|-0.02352150|0.06187285|0.07603523|0.01054182|-0.04044925|0.07349400|-
0.13836564|0.06418620|-0.03343910|0.09252818|0.01044974|-
0.10263962|0.00459680|0.03962868|0.08255708|0.04718511|-0.03129218|-
0.05016442|0.01958047|0.00264603|-0.03101716|-0.02905761|-0.01359326|-
0.03858718|-0.01572311|-0.02548057|0.05877890|-
0.04050557|0.03750503|0.06700855|0.05874502|0.07825154|-
0.00355426|0.01903197|0.00498147|-0.07881210|0.00000000|-0.05960625|-
0.04987202|-0.00138075|-0.03072614|-
0.04593481|0.00499481|0.03062444|0.04854004|0.04856795|-
0.04252950|0.06823035|0.04831520|0.00071800|-0.00733460|0.02326323|0.05408787|-
0.08365194|0.09335860|-0.09135234|0.01044529|-
0.00913032|0.04800938|0.01843412|0.03026514|-
0.04466689|0.03554082|0.03925864|0.06506035|0.05343359|-
0.02605321|0.04805842|0.09994715|-0.03423294|0.01691519|0.00180983|-

```
0.03224233|0.08819725|0.03905803|0.03997875|-0.04852342|0.05556690|0.02543087|-
0.00488903|0.08373825|-0.01088851|-0.03461628|0.07746255|-
0.03100988|0.00036735|-0.02162998|0.05038891|-
0.08484641|0.04023287|0.03999207|0.07557032|0.01878754|0.08266060|-
0.00024636|0.03139161|0.01360509|0.02633853|0.06425305|-
0.00337980|0.08635494|0.06564805|-0.02983328|-0.04299083|-0.01639626|-
0.00310801|-0.03284500|0.00256830|-0.04066422|0.04192689|0.01595493|0.06779454|-
0.03554782|0.00014695|0.04756820|-0.04636275|0.04779746|-0.01160627|0.00513149|-
0.02277845|-0.07837527|0.02301399|-0.06108354|0.09649085|0.18262914|-
0.01292571|-0.02849181|-0.02135289|0.11947743|-0.04465365|0.06056675|-
0.07236785|-0.00000002|0.04748648|-0.06727210|-0.07443058|-
0.01401228|0.02432247|0.08224294|-0.01993572|0.02343383|-0.02883206|-
0.02288137|0.03344429|0.03447076|-0.01500412|0.01586840|0.02013937|-0.04136627|-
0.00354916|-0.00977019|0.00399868|-0.02987850|0.02298432|0.00611404|-
0.00480101|-0.08306893|-0.05554646|-0.03697961|0.00840131|0.07042710|-
0.03862437|-0.03280582|0.04463454|0.12297195|0.00985497|-
0.03571197|0.03469852|0.05390766|-0.00655390|0.08150657|-0.03881459|-
0.00654811|0.04464525|0.01741038|-0.02537426|-0.06226552|0.01309725|0.00960539|-
0.03949535|-0.04986169|-0.01106449|-0.05871974|-
0.05512214|0.00796143|0.09373861|0.02742102|-0.05682594|0.02793728|-
0.04858829|0.06659685|-0.01170968|0.10258575|0.02004162|-0.02124332|-
0.05044460|-0.05105339']}, {'@type': 'g:T', '@value': 'id'}, 'addr1:315.0',
{'@type': 'g:T', '@value': 'label'}, 'Address1']}]}, 'meta': {'@type': 'g:Map',
'@value': []}}}
```

[ ]: