**Amazon Neptune**   
**to OpenSearch for implementing**   
**Full-Text Search**

# **Table of Contents**

About 3

Pre-requisites 3

Architecture 4

Working 4

Setup 5

Sample Output 5

Parameters in Template 6

Resources in Template 10

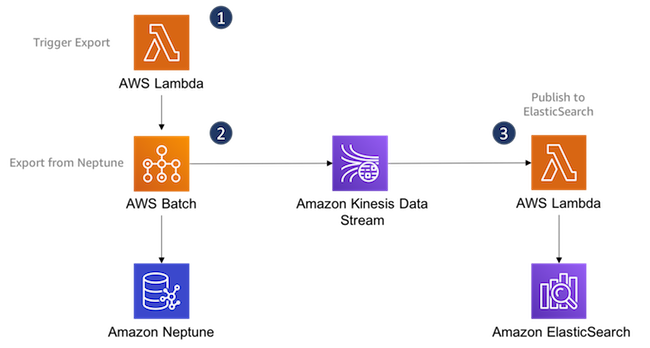
Monitoring and Debugging 10

**About:**

This solution allows you to index existing data in an Amazon Neptune database in OpenSearch *before* enabling Neptune's full-text search integration.

**Pre-requisites:**

* You have an existing Neptune cluster and an existing OpenSearch cluster in the same VPC
* OpenSearch version is less than 2.3
* You have VPC security groups that can be used to access your Neptune and OpenSearch clusters.
* Recommend using *this* against a static version of your data. Either suspend writes to your database while the export is taking place or run the export against a [snapshot](https://docs.aws.amazon.com/neptune/latest/userguide/backup-restore-create-snapshot.html) or [clone](https://docs.aws.amazon.com/neptune/latest/userguide/manage-console-cloning.html) of your database. The CloudFormation template allows you to specify whether you want the export process to clone your cluster or not.
* Export process uses SSL to connect to Neptune. It currently supports IAM Database Authentication for Gremlin, but not SPARQL.

**Architecture:**  


**Working:**

1. You trigger the export process via an AWS Lambda Function
2. The export process uses AWS Batch to host and execute *neptune-export*, which exports data from Neptune and publishes it to an Amazon Kinesis Data Stream in the [Neptune Streams format](https://docs.aws.amazon.com/neptune/latest/userguide/streams-change-formats.html).
3. A second AWS Lambda function polls the Kinesis Stream and publishes records to your Amazon ElasticSearch cluster. This function uses the same parsing and publishing code as the Neptune Streams ElasticSearch integration solution.

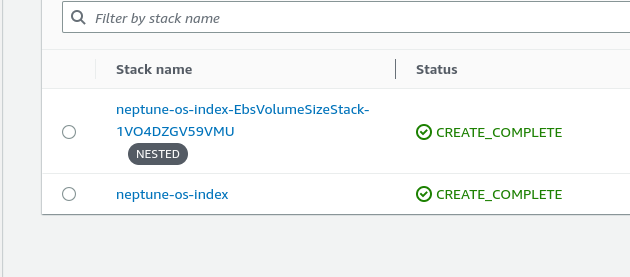
**Setup:**

1. Launch the CloudFormation Stack.
2. Invoke the first lambda.  
   aws lambda invoke \  
    --function-name arn:aws:lambda:eu-west-1:000000000000:function:export-neptune-to-kinesis-xxxx \  
    --region eu-west-1 \  
    /dev/stdout

The function returns the name and ID of an AWS Batch job that begins the export from Neptune.

1. Data should have been successfully populated in OpenSearch.

**Sample Output:**



**Parameters in template:**

1. Additional Params:

|  |  |
| --- | --- |
| Type | String |
| Description | Allows you to specify additional parameters to neptune-export, e.g. '--max-content-length 131072'. See <https://github.com/awslabs/amazon-neptune-tools/tree/master/neptune-export> for documentation of neptune-export params. |
| Default | “” |

1. BatchSize:

|  |  |
| --- | --- |
| Type | Number |
| Description | Number of records to retrieve from Kinesis Stream in a batch. (A single property graph Kinesis Stream record may contain multiple Neptune Streams records). |
| Default | 100 |
| MinValue | 1 |
| MaxValue | 10000 |

1. Clone Cluster:

|  |  |
| --- | --- |
| Type | String |
| Description | Clone Neptune cluster and export from clone (recommended for consistent exports that don't impact source cluster). |
| Default | false |

1. Concurrency:

|  |  |
| --- | --- |
| Type | Number |
| Description | Determines number of client threads in neptune-export |
| Default | 2 |
| MinValue | 1 |
| MaxValue | 8 |

1. ElasticSearchClientSecurityGroup:

|  |  |
| --- | --- |
| Type | String |
| Description | A VPC security group that AWS lambda can use to access the ElasticSearch cluster in your VPC. |

1. ElasticSearchEndpoint:

|  |  |
| --- | --- |
| Type | String |
| Description | ElasticSearch VPC endpoint (without 'https://' prefix). E.g. vpc-neptunestream.us-east-1.es.amazonaws.com |

1. ExportScope:

|  |  |
| --- | --- |
| Type | String |
| Description | Determines whether to export both nodes and edges, or nodes or edges only (Gremlin engine only). |
| Default | all |

1. GeoLocationFields:

|  |  |
| --- | --- |
| Type | String |
| Description | Determines whether to export both nodes and edges, or nodes or edges only (Gremlin engine only). |
| Default | all |

1. KeyPairName

|  |  |
| --- | --- |
| Type | AWS::EC2::KeyPair::KeyName |
| Description | Name of an existing EC2 KeyPair to enable SSH access to the export EC2 instance. |

1. KinesisShardCount

|  |  |
| --- | --- |
| Type | Number |
| Description | Determines number of shards in the Kinesis Stream and degree of parallelism when importing to ElasticSearch. |
| Default | 8 |

1. NeptuneClientSecurityGroup

|  |  |
| --- | --- |
| Type | String |
| Description | A VPC security group that neptune-export can use to access the Neptune cluster in your VPC. |

1. NeptuneEndpoint

|  |  |
| --- | --- |
| Type | String |
| Description | Neptune endpoint. E.g. mycluster-xxxxyyyy.cluster-yyyyzzzz.us-east-1.neptune.amazonaws.com |

1. NeptuneEngine

|  |  |
| --- | --- |
| Type | String |
| Description | Neptune engine. |

1. NeptunePort

|  |  |
| --- | --- |
| Type | String |
| Description | Neptune Port |

1. NumberOfReplica

|  |  |
| --- | --- |
| Type | Number |
| Description | Number of replicas for ElasticSearch index. Default value is 1. |

1. NumberOfShards

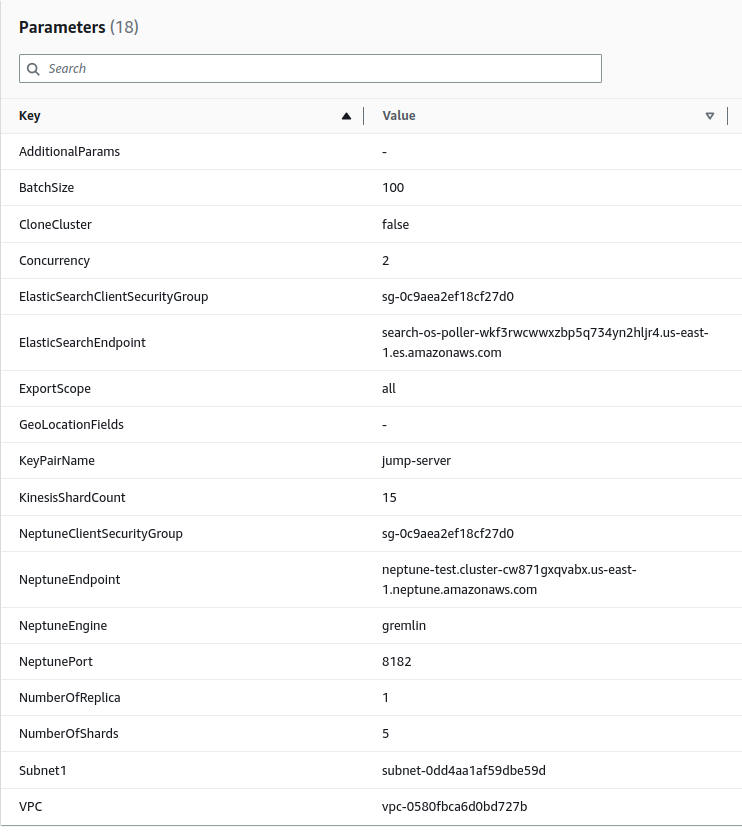
|  |  |
| --- | --- |
| Type | Number |
| Description | Number of shards for ElasticSearch index. Default value is 1. |

1. Subnet1

|  |  |
| --- | --- |
| Type | AWS::EC2::Subnet::Id |
| Description | Subnet in your Neptune/ElasticSearch VPC with route to the internet. |

1. VPC

|  |  |
| --- | --- |
| Type | AWS::EC2::VPC::Id |
| Description | ID of the VPC containing your Neptune and ElasticSearch clusters. |



**Resources in template:**

|  |  |
| --- | --- |
| **Logical ID** | **Type** |
| BatchAllowAllSecurityGroupIngress | AWS::EC2::SecurityGroupIngress |
| BatchExecutionRole | AWS::IAM::Role |
| BatchJobRole | AWS::IAM::Role |
| BatchSelfReferencingSecurityGroup | AWS::EC2::SecurityGroup |
| BatchSelfReferencingSecurityGroupIngress | AWS::EC2::SecurityGroupIngress |
| ComputeEnvironment | AWS::Batch::ComputeEnvironment |
| EbsVolumeSizeStack | AWS::CloudFormation::Stack |
| ECSRole | AWS::IAM::Role |
| ECSRoleInstanceProfile | AWS::IAM::InstanceProfile |
| EventSourceMapping | AWS::Lambda::EventSourceMapping |
| JobDefinition | AWS::Batch::JobDefinition |
| JobQueue | AWS::Batch::JobQueue |
| KinesisStream | AWS::Kinesis::Stream |
| KinesisToElasticSearchLambda | AWS::Lambda::Function |
| LambdaExecutionRole | AWS::IAM::Role |
| LambdaSelfReferencingSecurityGroup | AWS::EC2::SecurityGroup |
| LambdaSelfReferencingSecurityGroupIngress | AWS::EC2::SecurityGroupIngress |
| LaunchTemplate | AWS::EC2::LaunchTemplate |
| NeptuneExportLambda | AWS::Lambda::Function |
| NeptuneStreamsLayer | AWS::Lambda::LayerVersion |
| NeptuneToElasticSearchLayer | AWS::Lambda::LayerVersion |

**Monitoring and Debugging:**

To diagnose issues with the export from Neptune to Kinesis, consult the Amazon CloudWatch logs for your AWS Batch **export-neptune-to-kinesis-job**. These logs will indicate whether *neptune-export* was successfully downloaded to the batch instance, and the progress of the export job. When reviewing the logs, ensure that:

* *neptune-export* has been successfully downloaded to the Batch compute instance
* *neptune-export* has successfully exported nodes and relationships from Neptune and published them to Kinesis