1. **Create cluster**

* Enter cluster name
* Choose a cluster service role
* Ensure that the **AmazonEKSClusterPolicy** managed policy is attached to the role



* If you don’t have a role with AmazoneEKSClusterPolicy, read the link to create a new one: https://docs.aws.amazon.com/eks/latest/userguide/service\_IAM\_role.html

A screenshot of a computer

Description automatically generated

* Other things remain default, you can add a tag if you want to

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

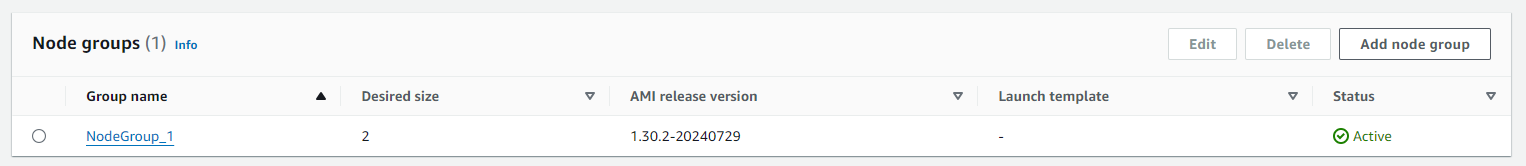
A screenshot of a computer

Description automatically generated

1. **Create node for cluster**

A screenshot of a computer

Description automatically generated



* Enter name for node group and choose role
* Role for node group must as least has these three:

A screenshot of a computer

Description automatically generated

* If you don’t have a role with AmazoneEKSClusterPolicy, reading the link to create a new one: https://docs.aws.amazon.com/eks/latest/userguide/create-node-role.html

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Choose machine OS and instances (CPUs, RAM, Network, …)

A screenshot of a computer

Description automatically generated

* Decide number of nodes in the cluster
* Desired size
* Minimum size
* Maximum size

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Configure to connect to eks cluster**
   1. **Configure the AWS CLI**

* First, we need to create an access key
* **A screenshot of a computer

  Description automatically generated**Remember to save your access key and secret access key

Figure. Create access and secret access key

* Command: aws configure
* Enter the Access key and secret key you just created
* Enter your region
* Output format will be **json**

A screen shot of a computer code

Description automatically generated

Figure. Configure aws key

* 1. **Authenticate MFA CLI**
     1. **Get session token**
* Command

|  |
| --- |
| **aws sts get-session-token**  **--serial-number arn-of-the-mfa-device**  **--token-code code-from-MFA-app** |

* A screenshot of a computer

  Description automatically generated**--serial-number:** Find it on the website
* **-–token-code:** Get the token from your MFA authenticator
* **Example**:

|  |
| --- |
| aws sts get-session-token  --serial-number arn:aws:iam::666243375423:mfa/tienpta3\_2  --token-code 123456 |

A computer screen shot of a black screen

Description automatically generated

Figure. Output after running the get session command

For more information: https://repost.aws/knowledge-center/authenticate-mfa-cli

* + 1. **Configure session token**
* Copy **AccessKeyId**, **SecretAccessKey**, **SessionToken** you just got from the previous step to the **./aws/credentials** file

A black screen with white text

Description automatically generated with medium confidence

Figure. Configure key and token

* + 1. **Update kube’s config to connect to the cluster**
* Command

|  |
| --- |
| aws eks  --region <region>  --name <cluster\_name>  update-kubeconfig  Example: aws eks --region ap-southeast-1 update-kubeconfig --name rag-cluster |



Figure. Update kubectl config

NOTE: The session token will expire after a specific time. You will need to create a new session token if needed

* Delete the old session token in the **./aws/credentials file**
* Replace the current access key and secret access key with the keys you create in step 3.1
* Then you follow 3.2.1 and 3.2.2

1. **Docker**

* Dockfile for frontend

**A screenshot of a computer program

Description automatically generated**

Figure. Frontend’s docker file

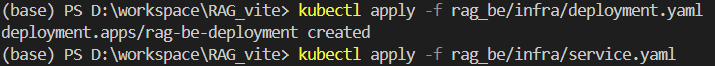
* Docker file for backend

**A screenshot of a computer program

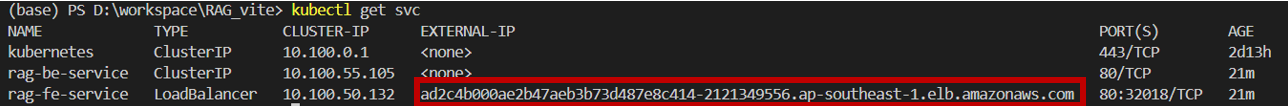
Description automatically generated**

Figure. Backend’s docker file

1. **Deployment**
   1. **Deploy backend**

* Deploy deployment and service
  1. **Deploy frontend**
* A black background with white text

  Description automatically generatedDeploy deployment service and configmap
  1. A screen shot of a computer

     Description automatically generated**Waiting for pods**
  2. **Service external-IP**
* After everything is running, you can access the app through the external-IP of the service of frontend

1. **Create a domain**

* A screenshot of a computer

  Description automatically generatedEnter subdomain name
* A screenshot of a computer

  Description automatically generatedChoose alias
* Choose endpoint: Alias to Application and Classic Load Balancer
* Choose region
* A screenshot of a computer

  Description automatically generatedChoose load balance

1. **Deploy Neo4j on Kubernetes pod (minikube)**
   1. **Export data**

Since data is stored in neo4j aura (online), we need to export the data and later import it into neo4j on k8s

* Edit **./Neo4jDesktop/relate-data/dbmss/<db\_name>/conf/apoc.conf** with content **apoc.export.file.enabled=true**

A screenshot of a computer

Description automatically generated

Figure. apoc.conf file

* Open the neo4j desktop, connect to the database, and run the command:

|  |
| --- |
| **CALL apoc.export.csv.all("<file\_name>.csv", {})** |

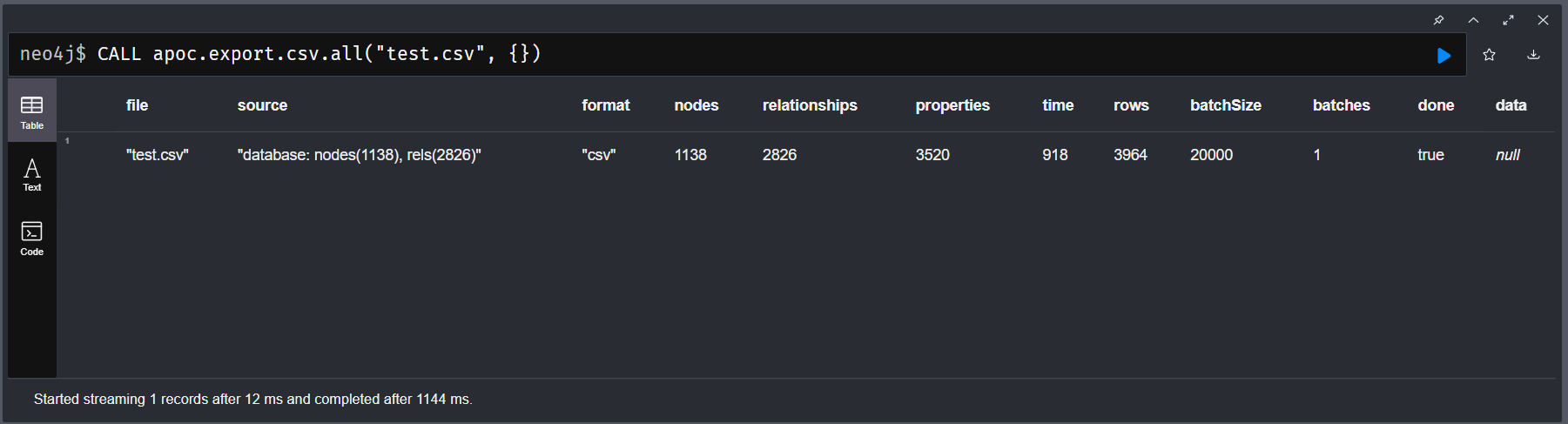
* ****The exported data will be saved in a csv file **./Neo4jDesktop/relate-data/dbmss/<db\_name>/import/<file\_name>.csv**

Figure. Output after exporting data

* 1. **Deploy Neo4j**
* Install helm if you don’t have (<https://helm.sh/docs/intro/install/>)
* Add the Neo4j Helm chart repository.

|  |
| --- |
| **helm repo add neo4j https://helm.neo4j.com/neo4j** |

* Update the repository.

|  |
| --- |
| **helm repo update** |

* Install Neo4j instance

|  |
| --- |
| **helm install my-neo4j-release neo4j/neo4j -f values.yaml** |

**A screen shot of a computer

Description automatically generated**

Figure. Yaml files for neo4j pod

**A screenshot of a computer program

Description automatically generated**Figure. Output after installing the neo4j instance

For more information:<https://neo4j.com/docs/operations-manual/current/kubernetes/>

* 1. **Migrate data**
* The export cvs file contains two main things: The node information and the relationship between nodes. You need to separate these into 2 files node.csv and relationship.csv respectively
* Preprocess before importing data
* change header **\_start**, \_**end**, **\_type** into **:START\_ID**, **:END\_ID**, **:TYPE** in relationship.csv

**A screenshot of a table

Description automatically generated**

Figure. Relationship file after preprocessing

* + Change header: **\_id** and **\_labels** into **:ID** and **:LABEL**
  + Remove all the colons at the beginning of each row in :LABEL
  + Replace the colon with the separate labels into semicolons

**A screenshot of a computer

Description automatically generated**

Figure. Node file after preprocessing

* Copy CSV files (node.csv and relationship.csv) to neo4j pod

|  |
| --- |
| Command:  **kubectl cp <db\_csv> <namespace>/<pod\_name>:/import/<file\_name>**  Example:  kubectl cp node.csv default:my-neo4j-release-0:/import/node.csv |

* Import data
* Stop the neo4j server before importing data

|  |
| --- |
| kubectl exec -it my-neo4j-0 -- /bin/sh  neo4j stop |

A black background with white text

Description automatically generated

Figure. Stopping neo4j pod

* A screenshot of a computer screen

  Description automatically generatedImport data using the following command

|  |
| --- |
| kubectl exec -it my-neo4j-0 -- /bin/sh  neo4j-admin database import full  --nodes=/import/node.csv  --relationships=/import/relationship.csv  --overwrite-destination  --multiline-fields=true |

Figure. Output after importing data