#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Neo4j configuration

#

# For more details and a complete list of settings, please see

# https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Paths of directories in the installation.

#server.directories.data=data

#server.directories.plugins=plugins

#server.directories.logs=logs

#server.directories.lib=lib

#server.directories.run=run

#server.directories.licenses=licenses

#server.directories.metrics=metrics

#server.directories.dumps.root=data/dumps

#server.directories.transaction.logs.root=data/transactions

# This setting constrains all `LOAD CSV` import files to be under the `import` directory. Remove or comment it out to

# allow files to be loaded from anywhere in the filesystem; this introduces possible security problems. See the

# `LOAD CSV` section of the manual for details.

server.directories.import=import

# Whether requests to Neo4j are authenticated.

# To disable authentication, uncomment this line

dbms.security.auth\_enabled=true

# Number of databases in Neo4j is limited.

# To change this limit please uncomment and adapt following setting:

# dbms.max\_databases=100

# Enable online backups to be taken from this database.

#server.backup.enabled=true

# By default the backup service will only listen on localhost.

# To enable remote backups you will have to bind to an external

# network interface (e.g. 0.0.0.0 for all interfaces).

# The protocol running varies depending on deployment. In a cluster this is the

# same protocol that runs on server.cluster.listen\_address.

#server.backup.listen\_address=0.0.0.0:6362

# Anonymous usage data reporting

# To disable, uncomment this line

#dbms.usage\_report.enabled=false

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Initial DBMS Settings

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Initial DBMS settings are picked up from the config file once, when a cluster first starts, and then transferred into

# the running DBMS. This means later changes to the values will not be seen. There are procedures to change the values

# after the initial start

# Name of the default database (aliases are not supported). Can be changed with the 'dbms.setDefaultDatabase' procedure.

#initial.dbms.default\_database=neo4j

# Initial default number of primary and secondary instances of user databases. If the user does not specify the number

# of primaries and secondaries in 'CREATE DATABASE', these values will be used, unless they are overwritten with the

# 'dbms.setDefaultAllocationNumbers' procedure.

#initial.dbms.default\_primaries\_count=1

#initial.dbms.default\_secondaries\_count=0

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Memory Settings

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#

# Memory settings are specified kibibytes with the 'k' suffix, mebibytes with

# 'm' and gibibytes with 'g'.

# If Neo4j is running on a dedicated server, then it is generally recommended

# to leave about 2-4 gigabytes for the operating system, give the JVM enough

# heap to hold all your transaction state and query context, and then leave the

# rest for the page cache.

# Java Heap Size: by default the Java heap size is dynamically calculated based

# on available system resources. Uncomment these lines to set specific initial

# and maximum heap size.

#server.memory.heap.initial\_size=512m

#server.memory.heap.max\_size=512m

# The amount of memory to use for mapping the store files.

# The default page cache memory assumes the machine is dedicated to running

# Neo4j, and is heuristically set to 50% of RAM minus the Java heap size.

#server.memory.pagecache.size=10g

# Limit the amount of memory that all of the running transaction can consume.

# The default value is 70% of the heap size limit.

#dbms.memory.transaction.total.max=256m

# Limit the amount of memory that a single transaction can consume.

# By default there is no limit.

#db.memory.transaction.max=16m

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Network connector configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# With default configuration Neo4j only accepts local connections.

# Use 0.0.0.0 to bind to all network interfaces on the machine. If you want to only use a specific interface

# (such as a private IP address on AWS, for example) then use that IP address instead.

#server.default\_listen\_address=0.0.0.0

# You can also choose a specific network interface, and configure a non-default

# port for each connector, by setting their individual listen\_address.

# The address at which this server can be reached by its clients. This may be the server's IP address or DNS name, or

# it may be the address of a reverse proxy which sits in front of the server. This setting may be overridden for

# individual connectors below.

#server.default\_advertised\_address=localhost

# You can also choose a specific advertised hostname or IP address, and

# configure an advertised port for each connector, by setting their

# individual advertised\_address.

# By default, encryption is turned off.

# To turn on encryption, an ssl policy for the connector needs to be configured

# Read more in SSL policy section in this file for how to define a SSL policy.

# Bolt connector

server.bolt.enabled=true

#server.bolt.tls\_level=DISABLED

#server.bolt.listen\_address=:7687

#server.bolt.advertised\_address=:7687

# HTTP Connector. There can be zero or one HTTP connectors.

server.http.enabled=true

#server.http.listen\_address=:7474

#server.http.advertised\_address=:7474

# HTTPS Connector. There can be zero or one HTTPS connectors.

server.https.enabled=false

#server.https.listen\_address=:7473

#server.https.advertised\_address=:7473

# Number of Neo4j worker threads.

#server.threads.worker\_count

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# SSL policy configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Each policy is configured under a separate namespace, e.g.

# dbms.ssl.policy.<scope>.\*

# <scope> can be any of 'bolt', 'https', 'cluster' or 'backup'

#

# The scope is the name of the component where the policy will be used

# Each component where the use of an ssl policy is desired needs to declare at least one setting of the policy.

# Allowable values are 'bolt', 'https', 'cluster' or 'backup'.

# E.g if bolt and https connectors should use the same policy, the following could be declared

# dbms.ssl.policy.bolt.base\_directory=certificates/default

# dbms.ssl.policy.https.base\_directory=certificates/default

# However, it's strongly encouraged to not use the same key pair for multiple scopes.

#

# N.B: Note that a connector must be configured to support/require

# SSL/TLS for the policy to actually be utilized.

#

# see: dbms.connector.\*.tls\_level

# SSL settings (dbms.ssl.policy.<scope>.\*)

# .base\_directory Base directory for SSL policies paths. All relative paths within the

# SSL configuration will be resolved from the base dir.

#

# .private\_key A path to the key file relative to the '.base\_directory'.

#

# .private\_key\_password The password for the private key.

#

# .public\_certificate A path to the public certificate file relative to the '.base\_directory'.

#

# .trusted\_dir A path to a directory containing trusted certificates.

#

# .revoked\_dir Path to the directory with Certificate Revocation Lists (CRLs).

#

# .verify\_hostname If true, the server will verify the hostname that the client uses to connect with. In order

# for this to work, the server public certificate must have a valid CN and/or matching

# Subject Alternative Names.

#

# .client\_auth How the client should be authorized. Possible values are: 'none', 'optional', 'require'.

#

# .tls\_versions A comma-separated list of allowed TLS versions. By default only TLSv1.2 and TLSv1.3 are allowed.

#

# .trust\_all Setting this to 'true' will ignore the trust truststore, trusting all clients and servers.

# Use of this mode is discouraged. It would offer encryption but no security.

#

# .ciphers A comma-separated list of allowed ciphers. The default ciphers are the defaults of

# the JVM platform.

# Bolt SSL configuration

#dbms.ssl.policy.bolt.enabled=true

#dbms.ssl.policy.bolt.base\_directory=certificates/bolt

#dbms.ssl.policy.bolt.private\_key=private.key

#dbms.ssl.policy.bolt.public\_certificate=public.crt

#dbms.ssl.policy.bolt.client\_auth=NONE

# Https SSL configuration

#dbms.ssl.policy.https.enabled=true

#dbms.ssl.policy.https.base\_directory=certificates/https

#dbms.ssl.policy.https.private\_key=private.key

#dbms.ssl.policy.https.public\_certificate=public.crt

#dbms.ssl.policy.https.client\_auth=NONE

# Cluster SSL configuration

#dbms.ssl.policy.cluster.enabled=true

#dbms.ssl.policy.cluster.base\_directory=certificates/cluster

#dbms.ssl.policy.cluster.private\_key=private.key

#dbms.ssl.policy.cluster.public\_certificate=public.crt

# Backup SSL configuration

#dbms.ssl.policy.backup.enabled=true

#dbms.ssl.policy.backup.base\_directory=certificates/backup

#dbms.ssl.policy.backup.private\_key=private.key

#dbms.ssl.policy.backup.public\_certificate=public.crt

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Logging configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# To enable HTTP logging, uncomment this line

#dbms.logs.http.enabled=true

# To enable GC Logging, uncomment this line

#server.logs.gc.enabled=true

# GC Logging Options

# see https://docs.oracle.com/en/java/javase/11/tools/java.html#GUID-BE93ABDC-999C-4CB5-A88B-1994AAAC74D5

#server.logs.gc.options=-Xlog:gc\*,safepoint,age\*=trace

# Number of GC logs to keep.

#server.logs.gc.rotation.keep\_number=5

# Size of each GC log that is kept.

#server.logs.gc.rotation.size=20m

# Log executed queries. One of OFF, INFO and VERBOSE. INFO logs queries longer than a given threshold, VERBOSE logs start and end of all queries.

#db.logs.query.enabled=VERBOSE

# If the execution of query takes more time than this threshold, the query is logged. If set to zero then all queries

# are logged. Only used if `db.logs.query.enabled` is set to INFO

#db.logs.query.threshold=0

# Include parameters for the executed queries being logged (this is enabled by default).

#db.logs.query.parameter\_logging\_enabled=true

# The security log is always enabled when `dbms.security.auth\_enabled=true`, for addition

# configuration, look at $NEO4J\_HOME/conf/server-logs.xml

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Cluster Configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Uncomment and specify these lines for running Neo4j in a cluster.

# See the cluster documentation at https://neo4j.com/docs/ for details.

# A comma-separated list of endpoints which a server should contact in order to discover other cluster members. It must

# be in the host:port format. For each machine in the cluster, the address will usually be the public ip address of

# that machine. The port will be the value used in the setting "server.discovery.advertised\_address" of that server.

#dbms.cluster.discovery.endpoints=localhost:5000,localhost:5001,localhost:5002

# Host and port to bind the cluster member discovery management communication.

# This is the setting to add to the collection of addresses in dbms.cluster.discovery.endpoints.

#server.discovery.listen\_address=:5000

#server.discovery.advertised\_address=:5000

# Network interface and port for the transaction shipping server to listen on.

# Please note that it is also possible to run the backup client against this port so always limit access to it via the

# firewall and configure an ssl policy.

#server.cluster.listen\_address=:6000

#server.cluster.advertised\_address=:6000

# Network interface and port for the RAFT server to listen on.

#server.cluster.raft.listen\_address=:7000

#server.cluster.raft.advertised\_address=:7000

# Network interface and port for server-side routing within the cluster. This allows requests to be forwarded

# from one cluster member to another, if the requests can't be satisfied by the first member (e.g. write requests

# received by a non-leader).

#server.routing.listen\_address=:7688

#server.routing.advertised\_address=:7688

# List a set of names for groups to which this server should belong. This

# is a comma-separated list and names should only use alphanumericals

# and underscore. This can be used to identify groups of servers in the

# configuration for load balancing and replication policies.

#

# The main intention for this is to group servers, but it is possible to specify

# a unique identifier here as well which might be useful for troubleshooting

# or other special purposes.

#server.groups

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Initial Server Settings

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Initial server settings are used as the default values when enabling a server, but can be overridden by specifying

# options when calling ENABLE (relevant for servers in a cluster \*after\* those that form the initial cluster).

# Restrict the modes of database that can be hosted on this server

# Allowed values:

# PRIMARY - Host standalone databases, and members of the consensus quorum for a multi-primary database.

# SECONDARY - Only host read replicas, eventually-consistent read-only instances of databases.

# NONE - Can host any mode of database

#initial.server.mode\_constraint=NONE

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Cluster Load Balancing

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# N.B: Read the online documentation for a thorough explanation!

# Selects the load balancing plugin that shall be enabled.

#dbms.routing.load\_balancing.plugin=server\_policies

####### Examples for "server\_policies" plugin #######

# Will select all available servers as the default policy, which is the

# policy used when the client does not specify a policy preference. The

# default configuration for the default policy is all().

#dbms.routing.load\_balancing.config.server\_policies.default=all()

# Will select servers in groups 'group1' or 'group2' under the default policy.

#dbms.routing.load\_balancing.config.server\_policies.default=groups(group1,group2)

# Slightly more advanced example:

# Will select servers in 'group1', 'group2' or 'group3', but only if there are at least 2.

# This policy will be exposed under the name of 'mypolicy'.

#dbms.routing.load\_balancing.config.server\_policies.mypolicy=groups(group1,group2,group3) -> min(2)

# Below will create an even more advanced policy named 'regionA' consisting of several rules

# yielding the following behaviour:

#

# select servers in regionA, if at least 2 are available

# otherwise: select servers in regionA and regionB, if at least 2 are available

# otherwise: select all servers

#

# The intention is to create a policy for a particular region which prefers

# a certain set of local servers, but which will fallback to other regions

# or all available servers as required.

#

# N.B: The following configuration uses the line-continuation character \

# which allows you to construct an easily readable rule set spanning

# several lines.

#

#dbms.routing.load\_balancing.config.server\_policies.policyA=\

#groups(regionA) -> min(2);\

#groups(regionA,regionB) -> min(2);

# Note that implicitly the last fallback is to always consider all() servers,

# but this can be prevented by specifying a halt() as the last rule.

#

#dbms.routing.load\_balancing.config.server\_policies.regionA\_only=\

#groups(regionA);\

#halt();

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Cluster Additional Configuration Options

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The following settings are used less frequently.

# If you don't know what these are, you don't need to change these from their default values.

# Cluster Routing Connector. Disable the opening of an additional port to allow

# for internal communication using the same security configuration as CLUSTER

#dbms.routing.enabled=false

# The time window within which the loss of the leader is detected and the first re-election attempt is held.

# The window should be significantly larger than typical communication delays to make conflicts unlikely.

#dbms.cluster.raft.leader\_failure\_detection\_window=20s-23s

# The rate at which leader elections happen. Note that due to election conflicts it might take several attempts to

# find a leader. The window should be significantly larger than typical communication delays to make conflicts unlikely.

#dbms.cluster.raft.election\_failure\_detection\_window=3s-6s

# The time limit allowed for a new member to attempt to update its data to match the rest of the cluster.

#dbms.cluster.raft.membership.join\_timeout=10m

# Maximum amount of lag accepted for a new follower to join the Raft group.

#dbms.cluster.raft.membership.join\_max\_lag=10s

# Raft log pruning frequency.

#dbms.cluster.raft.log.pruning\_frequency=10m

# The size to allow the raft log to grow before rotating.

#dbms.cluster.raft.log.rotation\_size=250M

# The name of a server\_group whose members should be prioritized as leaders for the given database.

# This does not guarantee that members of this group will be leader at all times, but the cluster

# will attempt to transfer leadership to such a member when possible.

# N.B. the final portion of this config key is dynamic and refers to the name of the database being configured.

# You may specify multiple `db.cluster.raft.leader\_transfer.priority\_group.<database-name>=<server-group>` pairs:

#db.cluster.raft.leader\_transfer.priority\_group.foo

#db.cluster.raft.leader\_transfer.priority\_group.neo4j

# Which strategy to use when transferring database leaderships around a cluster.

# This can be one of `equal\_balancing` or `no\_balancing`.

# `equal\_balancing` automatically ensures that each Core server holds the leader role for an equal number of databases.

# `no\_balancing` prevents any automatic balancing of the leader role.

# Note that if a `leadership\_priority\_group` is specified for a given database,

# the value of this setting will be ignored for that database.

#dbms.cluster.raft.leader\_transfer.balancing\_strategy=equal\_balancing

# The following setting controls how frequently a server hosting a secondary for a given database attempts to

# fetch an update from a server hosting a primary for that database

#db.cluster.catchup.pull\_interval=1s

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Security Configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The authentication and authorization providers that contains both users and roles.

# This can be one of the built-in `native` or `ldap` auth providers,

# or it can be an externally provided plugin, with a custom name prefixed by `plugin`,

# i.e. `plugin-<AUTH\_PROVIDER\_NAME>`.

#dbms.security.authentication\_providers=native

#dbms.security.authorization\_providers=native

# The time to live (TTL) for cached authentication and authorization info when using

# external auth providers (LDAP or plugin). Setting the TTL to 0 will

# disable auth caching.

#dbms.security.auth\_cache\_ttl=10m

# The maximum capacity for authentication and authorization caches (respectively).

#dbms.security.auth\_cache\_max\_capacity=10000

# Set to log successful authentication events to the security log.

# If this is set to `false` only failed authentication events will be logged, which

# could be useful if you find that the successful events spam the logs too much,

# and you do not require full auditing capability.

#dbms.security.log\_successful\_authentication=true

#================================================

# LDAP Auth Provider Configuration

#================================================

# URL of LDAP server to use for authentication and authorization.

# The format of the setting is `<protocol>://<hostname>:<port>`, where hostname is the only required field.

# The supported values for protocol are `ldap` (default) and `ldaps`.

# The default port for `ldap` is 389 and for `ldaps` 636.

# For example: `ldaps://ldap.example.com:10389`.

#

# NOTE: You may want to consider using STARTTLS (`dbms.security.ldap.use\_starttls`) instead of LDAPS

# for secure connections, in which case the correct protocol is `ldap`.

#dbms.security.ldap.host=localhost

# Use secure communication with the LDAP server using opportunistic TLS.

# First an initial insecure connection will be made with the LDAP server, and then a STARTTLS command

# will be issued to negotiate an upgrade of the connection to TLS before initiating authentication.

#dbms.security.ldap.use\_starttls=false

# The LDAP referral behavior when creating a connection. This is one of `follow`, `ignore` or `throw`.

# `follow` automatically follows any referrals

# `ignore` ignores any referrals

# `throw` throws an exception, which will lead to authentication failure

#dbms.security.ldap.referral=follow

# The timeout for establishing an LDAP connection. If a connection with the LDAP server cannot be

# established within the given time the attempt is aborted.

# A value of 0 means to use the network protocol's (i.e., TCP's) timeout value.

#dbms.security.ldap.connection\_timeout=30s

# The timeout for an LDAP read request (i.e. search). If the LDAP server does not respond within

# the given time the request will be aborted. A value of 0 means wait for a response indefinitely.

#dbms.security.ldap.read\_timeout=30s

#----------------------------------

# LDAP Authentication Configuration

#----------------------------------

# LDAP authentication mechanism. This is one of `simple` or a SASL mechanism supported by JNDI,

# for example `DIGEST-MD5`. `simple` is basic username

# and password authentication and SASL is used for more advanced mechanisms. See RFC 2251 LDAPv3

# documentation for more details.

#dbms.security.ldap.authentication.mechanism=simple

# LDAP user DN template. An LDAP object is referenced by its distinguished name (DN), and a user DN is

# an LDAP fully-qualified unique user identifier. This setting is used to generate an LDAP DN that

# conforms with the LDAP directory's schema from the user principal that is submitted with the

# authentication token when logging in.

# The special token {0} is a placeholder where the user principal will be substituted into the DN string.

#dbms.security.ldap.authentication.user\_dn\_template=uid={0},ou=users,dc=example,dc=com

# Determines if the result of authentication via the LDAP server should be cached or not.

# Caching is used to limit the number of LDAP requests that have to be made over the network

# for users that have already been authenticated successfully. A user can be authenticated against

# an existing cache entry (instead of via an LDAP server) as long as it is alive

# (see `dbms.security.auth\_cache\_ttl`).

# An important consequence of setting this to `true` is that

# Neo4j then needs to cache a hashed version of the credentials in order to perform credentials

# matching. This hashing is done using a cryptographic hash function together with a random salt.

# Preferably a conscious decision should be made if this method is considered acceptable by

# the security standards of the organization in which this Neo4j instance is deployed.

#dbms.security.ldap.authentication.cache\_enabled=true

#----------------------------------

# LDAP Authorization Configuration

#----------------------------------

# Authorization is performed by searching the directory for the groups that

# the user is a member of, and then map those groups to Neo4j roles.

# Perform LDAP search for authorization info using a system account instead of the user's own account.

#

# If this is set to `false` (default), the search for group membership will be performed

# directly after authentication using the LDAP context bound with the user's own account.

# The mapped roles will be cached for the duration of `dbms.security.auth\_cache\_ttl`,

# and then expire, requiring re-authentication. To avoid frequently having to re-authenticate

# sessions you may want to set a relatively long auth cache expiration time together with this option.

# NOTE: This option will only work if the users are permitted to search for their

# own group membership attributes in the directory.

#

# If this is set to `true`, the search will be performed using a special system account user

# with read access to all the users in the directory.

# You need to specify the username and password using the settings

# `dbms.security.ldap.authorization.system\_username` and

# `dbms.security.ldap.authorization.system\_password` with this option.

# Note that this account only needs read access to the relevant parts of the LDAP directory

# and does not need to have access rights to Neo4j, or any other systems.

#dbms.security.ldap.authorization.use\_system\_account=false

# An LDAP system account username to use for authorization searches when

# `dbms.security.ldap.authorization.use\_system\_account` is `true`.

# Note that the `dbms.security.ldap.authentication.user\_dn\_template` will not be applied to this username,

# so you may have to specify a full DN.

#dbms.security.ldap.authorization.system\_username

# An LDAP system account password to use for authorization searches when

# `dbms.security.ldap.authorization.use\_system\_account` is `true`.

#dbms.security.ldap.authorization.system\_password

# The name of the base object or named context to search for user objects when LDAP authorization is enabled.

# A common case is that this matches the last part of `dbms.security.ldap.authentication.user\_dn\_template`.

#dbms.security.ldap.authorization.user\_search\_base=ou=users,dc=example,dc=com

# The LDAP search filter to search for a user principal when LDAP authorization is

# enabled. The filter should contain the placeholder token {0} which will be substituted for the

# user principal.

#dbms.security.ldap.authorization.user\_search\_filter=(&(objectClass=\*)(uid={0}))

# A list of attribute names on a user object that contains groups to be used for mapping to roles

# when LDAP authorization is enabled. This setting is ignored when `dbms.ldap\_authorization\_nested\_groups\_enabled` is `true`.

#dbms.security.ldap.authorization.group\_membership\_attributes=memberOf

# This setting determines whether multiple LDAP search results will be processed (as is required for the lookup of nested groups).

# If set to `true` then instead of using attributes on the user object to determine group membership (as specified by

# `dbms.security.ldap.authorization.group\_membership\_attributes`), the `user` object will only be used to determine the user's

# Distinguished Name, which will subsequently be used with `dbms.security.ldap.authorization.user\_search\_filter`

# in order to perform a nested group search. The Distinguished Names of the resultant group search results will be used to determine roles.

#dbms.security.ldap.authorization.nested\_groups\_enabled=false

# The search template which will be used to find the nested groups which the user is a member of.

# The filter should contain the placeholder token `{0}` which will be substituted with the user's

# Distinguished Name (which is found for the specified user principle using `dbms.security.ldap.authorization.user\_search\_filter`).

# The default value specifies Active Directory's LDAP\_MATCHING\_RULE\_IN\_CHAIN (aka 1.2.840.113556.1.4.1941) implementation

# which will walk the ancestry of group membership for the specified user.

#dbms.security.ldap.authorization.nested\_groups\_search\_filter=(&(objectclass=group)(member:1.2.840.113556.1.4.1941:={0}))

# An authorization mapping from LDAP group names to Neo4j role names.

# The map should be formatted as a semicolon separated list of key-value pairs, where the

# key is the LDAP group name and the value is a comma separated list of corresponding role names.

# For example: group1=role1;group2=role2;group3=role3,role4,role5

#

# You could also use whitespaces and quotes around group names to make this mapping more readable,

# for example: dbms.security.ldap.authorization.group\_to\_role\_mapping=\

# "cn=Neo4j Read Only,cn=users,dc=example,dc=com" = reader; \

# "cn=Neo4j Read-Write,cn=users,dc=example,dc=com" = publisher; \

# "cn=Neo4j Schema Manager,cn=users,dc=example,dc=com" = architect; \

# "cn=Neo4j Administrator,cn=users,dc=example,dc=com" = admin

#dbms.security.ldap.authorization.group\_to\_role\_mapping

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# OpenID Connect configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The display name for the provider. This will be displayed in clients such as Neo4j Browser and Bloom.

#dbms.security.oidc.<provider>.display\_name

# The OIDC auth\_flow for clients such as Neo4j Browser and Bloom to use. Supported values are 'pkce' and 'implicit'

#dbms.security.oidc.<provider>.auth\_flow=pkce

# The OpenID Connect Discovery URL for the provider

#dbms.security.oidc.<provider>.well\_known\_discovery\_uri

# URL of the provider's Authorization Endpoint

#dbms.security.oidc.<provider>.auth\_endpoint

# Parameters to use with the Authorization Endpoint.

#dbms.security.oidc.<provider>.auth\_params

# URL of the provider's OAuth 2.0 Token Endpoint

#dbms.security.oidc.<provider>.token\_endpoint

# Parameters to use with the Token Endpoint.

#dbms.security.oidc.<provider>.token\_params

# URL of the provider's JSON Web Key Set

#dbms.security.oidc.<provider>.jwks\_uri

# URL of the provider's UserInfo Endpoint

#dbms.security.oidc.<provider>.user\_info\_uri

# URL that the provider asserts as its issuer identifier. This will be checked against the iss claim in the token

#dbms.security.oidc.<provider>.issuer

# The expected value for the `aud` claim

#dbms.security.oidc.<provider>.audience

# The client\_id of this client as issued by the provider.

#dbms.security.oidc.<provider>.client\_id

# Whether to fetch the groups claim from the user info endpoint on the identity provider. The default is false, read it from the token.

#dbms.security.oidc.<provider>.get\_groups\_from\_user\_info=false

# Whether to fetch the username claim from the user info endpoint on the identity provider. The default is false, read it from the token.

#dbms.security.oidc.<provider>.get\_username\_from\_user\_info=false

# The claim to use for the database username.

#dbms.security.oidc.<provider>.claims.username=sub

# The claim to use for the database roles.

#dbms.security.oidc.<provider>.claims.groups

# General parameters to use with the Identity Provider.

#dbms.security.oidc.<provider>.params

# General config to use with the Identity Provider.

#dbms.security.oidc.<provider>.config

# An authorization mapping from identity provider group names to Neo4j role names. See dbms.security.ldap.authorization.group\_to\_role\_mapping above

# for the format.

#dbms.security.oidc.<provider>.authorization.group\_to\_role\_mapping

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Miscellaneous configuration

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Compresses the metric archive files.

server.metrics.csv.rotation.compression=zip

# Determines if Cypher will allow using file URLs when loading data using

# `LOAD CSV`. Setting this value to `false` will cause Neo4j to fail `LOAD CSV`

# clauses that load data from the file system.

#dbms.security.allow\_csv\_import\_from\_file\_urls=true

# Value of the Access-Control-Allow-Origin header sent over any HTTP or HTTPS

# connector. This defaults to '\*', which allows broadest compatibility. Note

# that any URI provided here limits HTTP/HTTPS access to that URI only.

#dbms.security.http\_access\_control\_allow\_origin=\*

# Value of the HTTP Strict-Transport-Security (HSTS) response header. This header

# tells browsers that a webpage should only be accessed using HTTPS instead of HTTP.

# It is attached to every HTTPS response. Setting is not set by default so

# 'Strict-Transport-Security' header is not sent. Value is expected to contain

# directives like 'max-age', 'includeSubDomains' and 'preload'.

#dbms.security.http\_strict\_transport\_security

# Retention policy for transaction logs needed to perform recovery and backups.

db.tx\_log.rotation.retention\_policy=2 days 2G

# Limit the number of IOs the background checkpoint process will consume per second.

# This setting is advisory, is ignored in Neo4j Community Edition, and is followed to

# best effort in Enterprise Edition.

# An IO is in this case a 8 KiB (mostly sequential) write. Limiting the write IO in

# this way will leave more bandwidth in the IO subsystem to service random-read IOs,

# which is important for the response time of queries when the database cannot fit

# entirely in memory. The only drawback of this setting is that longer checkpoint times

# may lead to slightly longer recovery times in case of a database or system crash.

# A lower number means lower IO pressure, and consequently longer checkpoint times.

# Set this to -1 to disable the IOPS limit and remove the limitation entirely,

# this will let the checkpointer flush data as fast as the hardware will go.

# Removing the setting, or commenting it out, will set the default value of 600.

# db.checkpoint.iops.limit=600

# Whether or not any database on this instance are read\_only by default.

# If false, individual databases may be marked as read\_only using dbms.database.read\_only.

# If true, individual databases may be marked as writable using dbms.databases.writable.

#dbms.databases.default\_to\_read\_only=false

# Comma separated list of JAX-RS packages containing JAX-RS resources, one

# package name for each mountpoint. The listed package names will be loaded

# under the mountpoints specified. Uncomment this line to mount the

# org.neo4j.examples.server.unmanaged.HelloWorldResource.java from

# neo4j-server-examples under /examples/unmanaged, resulting in a final URL of

# http://localhost:7474/examples/unmanaged/helloworld/{nodeId}

#server.unmanaged\_extension\_classes=org.neo4j.examples.server.unmanaged=/examples/unmanaged

# A comma separated list of procedures and user defined functions that are allowed

# full access to the database through unsupported/insecure internal APIs.

#dbms.security.procedures.unrestricted=my.extensions.example,my.procedures.\*

# A comma separated list of procedures to be loaded by default.

# Leaving this unconfigured will load all procedures found.

#dbms.security.procedures.allowlist=apoc.coll.\*,apoc.load.\*,gds.\*

# For how long should drivers cache the discovery data from

# the dbms.routing.getRoutingTable() procedure. Defaults to 300s.

#dbms.routing\_ttl=300s

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# JVM Parameters

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# G1GC generally strikes a good balance between throughput and tail

# latency, without too much tuning.

server.jvm.additional=-XX:+UseG1GC

# Have common exceptions keep producing stack traces, so they can be

# debugged regardless of how often logs are rotated.

server.jvm.additional=-XX:-OmitStackTraceInFastThrow

# Make sure that `initmemory` is not only allocated, but committed to

# the process, before starting the database. This reduces memory

# fragmentation, increasing the effectiveness of transparent huge

# pages. It also reduces the possibility of seeing performance drop

# due to heap-growing GC events, where a decrease in available page

# cache leads to an increase in mean IO response time.

# Try reducing the heap memory, if this flag degrades performance.

server.jvm.additional=-XX:+AlwaysPreTouch

# Trust that non-static final fields are really final.

# This allows more optimizations and improves overall performance.

# NOTE: Disable this if you use embedded mode, or have extensions or dependencies that may use reflection or

# serialization to change the value of final fields!

server.jvm.additional=-XX:+UnlockExperimentalVMOptions

server.jvm.additional=-XX:+TrustFinalNonStaticFields

# Disable explicit garbage collection, which is occasionally invoked by the JDK itself.

server.jvm.additional=-XX:+DisableExplicitGC

# Allow Neo4j to use @Contended annotation

server.jvm.additional=-XX:-RestrictContended

# Restrict size of cached JDK buffers to 1 KB

server.jvm.additional=-Djdk.nio.maxCachedBufferSize=1024

# More efficient buffer allocation in Netty by allowing direct no cleaner buffers.

server.jvm.additional=-Dio.netty.tryReflectionSetAccessible=true

# Exits JVM on the first occurrence of an out-of-memory error. Its preferable to restart VM in case of out of memory errors.

# server.jvm.additional=-XX:+ExitOnOutOfMemoryError

# Expand Diffie Hellman (DH) key size from default 1024 to 2048 for DH-RSA cipher suites used in server TLS handshakes.

# This is to protect the server from any potential passive eavesdropping.

server.jvm.additional=-Djdk.tls.ephemeralDHKeySize=2048

# This mitigates a DDoS vector.

server.jvm.additional=-Djdk.tls.rejectClientInitiatedRenegotiation=true

# Enable remote debugging

#server.jvm.additional=-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=\*:5005

# This filter prevents deserialization of arbitrary objects via java object serialization, addressing potential vulnerabilities.

# By default this filter whitelists all neo4j classes, as well as classes from the hazelcast library and the java standard library.

# These defaults should only be modified by expert users!

# For more details (including filter syntax) see: https://openjdk.java.net/jeps/290

#server.jvm.additional=-Djdk.serialFilter=java.\*\*;org.neo4j.\*\*;com.neo4j.\*\*;com.hazelcast.\*\*;net.sf.ehcache.Element;com.sun.proxy.\*;org.openjdk.jmh.\*\*;!\*

# Increase the default flight recorder stack sampling depth from 64 to 256, to avoid truncating frames when profiling.

server.jvm.additional=-XX:FlightRecorderOptions=stackdepth=256

# Allow profilers to sample between safepoints. Without this, sampling profilers may produce less accurate results.

server.jvm.additional=-XX:+UnlockDiagnosticVMOptions

server.jvm.additional=-XX:+DebugNonSafepoints

# Open modules for neo4j to allow internal access

server.jvm.additional=--add-opens=java.base/java.nio=ALL-UNNAMED

server.jvm.additional=--add-opens=java.base/java.io=ALL-UNNAMED

server.jvm.additional=--add-opens=java.base/sun.nio.ch=ALL-UNNAMED

# Enable native memory access

server.jvm.additional=--enable-native-access=ALL-UNNAMED

# Enable access to JDK vector API

# server.jvm.additional=--add-modules=jdk.incubator.vector

# Disable logging JMX endpoint.

server.jvm.additional=-Dlog4j2.disable.jmx=true

# Limit JVM metaspace and code cache to allow garbage collection. Used by cypher for code generation and may grow indefinitely unless constrained.

# Useful for memory constrained environments

#server.jvm.additional=-XX:MaxMetaspaceSize=1024m

#server.jvm.additional=-XX:ReservedCodeCacheSize=512m

# Allow big methods to be JIT compiled.

# Useful for big queries and big expressions where cypher code generation can create large methods.

#server.jvm.additional=-XX:-DontCompileHugeMethods

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Wrapper Windows NT/2000/XP Service Properties

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# WARNING - Do not modify any of these properties when an application

# using this configuration file has been installed as a service.

# Please uninstall the service before modifying this section. The

# service can then be reinstalled.

# Name of the service

server.windows\_service\_name=neo4j

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Other Neo4j system properties

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dbms.memory.heap.initial\_size=512m

dbms.memory.heap.max\_size=1G

dbms.memory.pagecache.size=512m

dbms.windows\_service\_name=neo4j-relate-dbms-4cb9dd62-da06-4204-884e-6c87260a5332

dbms.jvm.additional=-Dlog4j2.formatMsgNoLookups=true

# Apoc added by me

dbms.security.procedures.unrestricted=apoc.\*

dbms.security.procedures.allowlist=apoc.\*