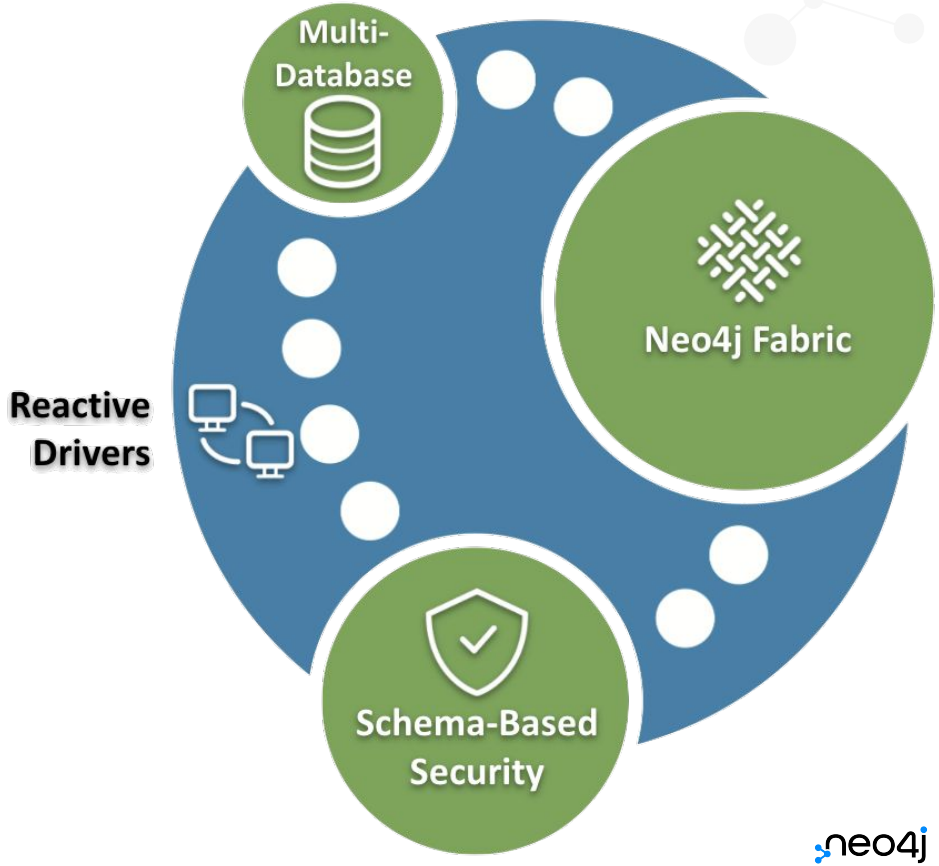
Best Practice & Considerations when leveraging Neo4j Multi Database Support

Neo4j Customer Success
June 2021

Neo4j 4.X in a nutshell

The Largest Investment in Graph Databases to date

- *A new architecture*
- *Secure*
- *Scalable*
- *Robust*
- *Easy to operate*



Support for Multiple Databases

- A DBMS (i.e. a Standalone Neo4j Server or Causal Cluster) can manage one or more databases
  - In Causal Cluster, databases are causally consistent
- Databases are structurally and logically separated:
  - Structurally: data is stored separately (but memory and CPU are shared)
  - Logically: relationships cannot be set between nodes in different databases
- The max number of databases is theoretically unlimited
  - Default: `dbms.max_databases = 100`
  - Tested up to 500 databases (4.2)
- Databases define a transaction and execution context
  - Transactions cannot span across databases
  - Procedures are executed “within” a database (although they can access data stored in other databases
- 3.X Compatibility: DBMS offers a default database:
  - The configuration parameter for the default database is `dbms.default_database = database-name`
  - In new installations, the initial default database is **`neo4j`**

Support for Multiple Databases

Use Cases:

- **Multi-tenancy**: A single instance of Neo4j Server or Neo4j Cluster may serve multiple customers/users within an organization.
- **Rapid Testing/Development/Deployment**: DBAs and developers create multiple copies of the same database.
- **Scalability**: Data is organized in physically separated structures.
- **Cloud-Friendly Elasticity:** Databases can be associated to Cloud storage and easily detached from a server and attached to another server.

The `system` database:

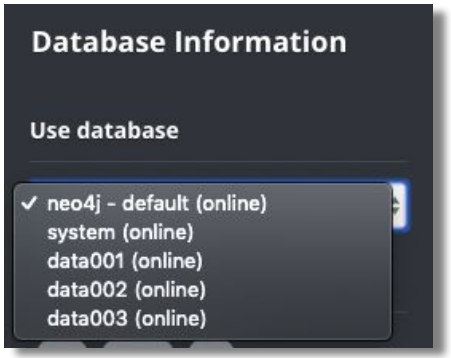
- Internal repository containing system information, available for the whole DBMS (standalone and cluster).

Administration commands:

- `CREATE | DROP | START | STOP DATABASE` *`name`*

User commands:

- HTTP API: `http://server:port/.../database`
- Browser & Cypher Shell: `:USE` *`database`*
- Drivers: `Session(`*`database`*`)`
- Browser:



Key Considerations

- Global Settings
  - Memory Settings
  - Transaction Limit Settings
  - File handles
  - Import Directory
  - Disk Space / Retention Settings
- Users/Roles/Privileges
- Backup / Restore
- Monitoring
- Leveraging Clustering Server Groups
- Onboarding a New Database

Global Settings

Global DBMS Settings

Key Settings that apply to entire Neo4j Instance (DBMS)

- These will be shared across all database instances
- Key Memory Settings
  - Pagecache
    `dbms.memory.pagecache.size`
    `Single memory area for all database and index pages`
  - Heap
    `dbms.memory.heap.initial_size`
    `Dbms.memory.heap.max_size`
    `Shared JVM Heap for all databases`
  - Off Heap
    `Dbms.tx_state.memory_allocation`
    `Shared Transaction Off Heap memory`
  - “Other” Memory
    `Share area for things like direct memory`
    KB - Understanding Memory Consumption

Recommendations

- Ideally, set pagecache size to sum of all running database sizes
- Monitor expected HEAP usage across databases and applications to set ideal heap size
- Set TX State allocation to ON_HEAP
- Leverage HEAP Memory Limits

Global DBMS Settings (continued)

Memory accounting: track java heap used by Neo4j

- Estimate memory used
- Configure thresholds:
  - Per DBMS
    `dbms.memory.transaction.global_max_size`
  - Per Database
    `dbms.memory.transaction.database_max_size`
  - Per Transaction
    `dbms.memory.transaction.max_size`
- Any query that brings memory used above threshold is killed
- Show memory used in PROFILE,
  `dbms.listTransactions(), dbms.listPools()`

- Avoid Out Of Memory errors
- Manage fairness across databases
- Manage fairness across transactions

Global DBMS Settings (continued)

Key Settings that apply to entire Neo4j Instance (DBMS)

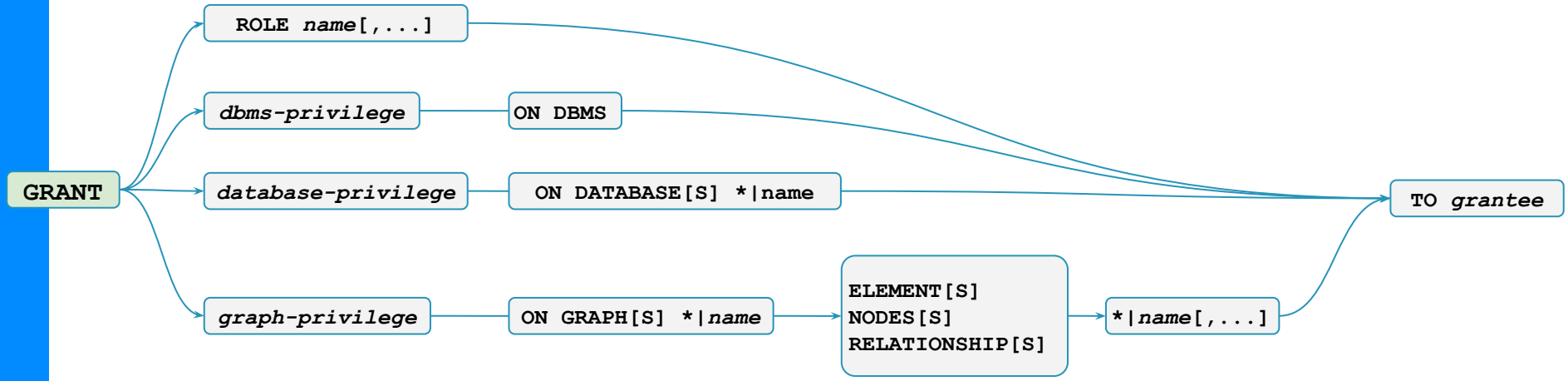
- These will be shared across all database instances
- File Handles
  - Number of open file handles per session
    `Single setting for the entire dbms session`
    `Previous recommendation was minimum 40000`
    Open Files Documentation
  - Import Directory
    `dbms.directories.import`
    `Shared folder for entire instance`
  - Disk Space
    `Every database will have their own files`
  - Retention Settings
    `Help with Disk space usage`
    Retention Settings

Recommendations

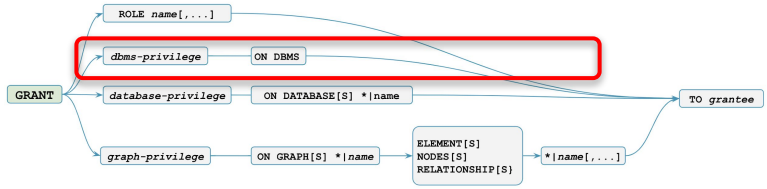
- Set the file limits high enough to account for all indexes and connections across the instance.
- Have a sub directory within the import folder for each database. Have process to create this when creating database.
- If a security concern, can leverage http/s for import location
- Make sure to account for more disk space that is used and monitor free space
- Set retention settings for Transaction logs, metrics (if enabled) and other logs

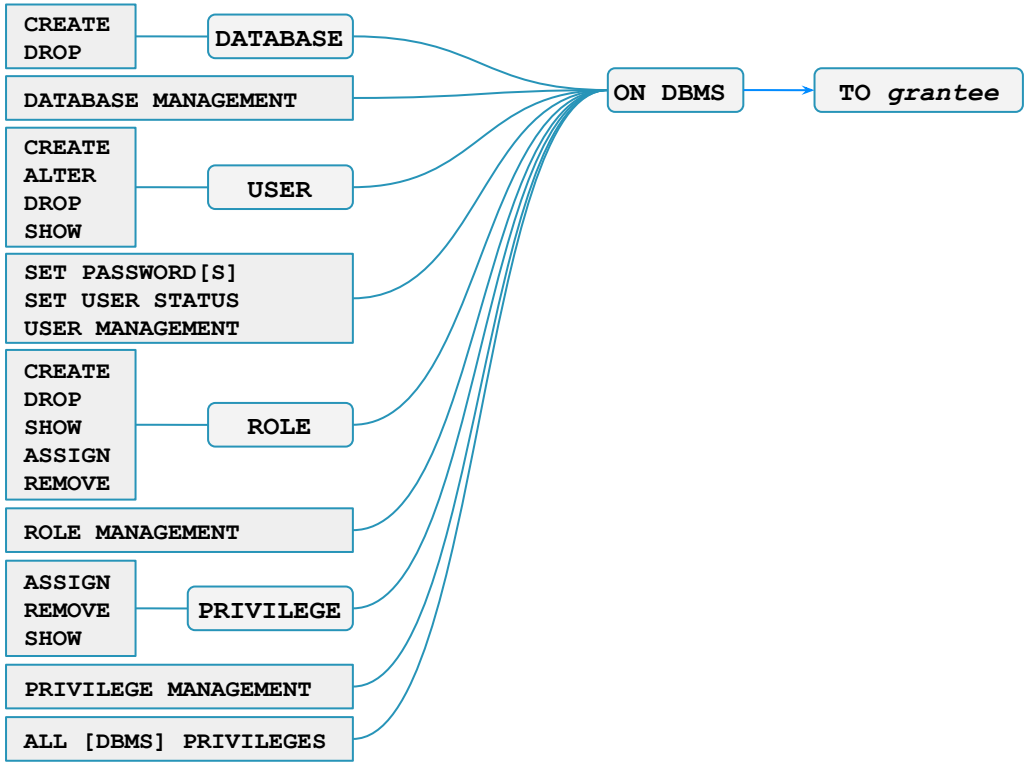
Users/Roles/Privileges

Schema-Based Security

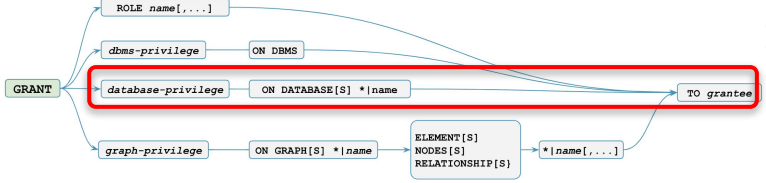


DBMS Privileges



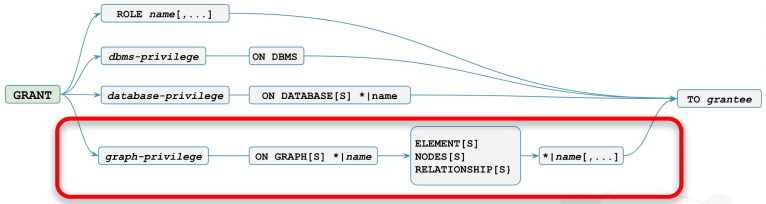


Database Privileges




```
ACCESS

START
STOP

CREATE INDEX[ES]
DROP INDEX[ES]
INDEX[ES] [MANAGEMENT]

CREATE CONSTRAINT[S]
DROP CONSTRAINT[S]
CONSTRAINT[S] [MANAGEMENT]

CREATE NEW [NODE] LABEL[S]
CREATE NEW [RELATIONSHIP] TYPE[S]
CREATE NEW [PROPERTY] NAME[S]
NAME [MANAGEMENT]

ALL [[DATABASE] PRIVILEGES]

SHOW TRANSACTION[S]
TERMINATE TRANSACTION[S]
TRANSACTION [MANAGEMENT]

(*|user[,...])

ON
DEFAULT DATABASE
DATABASE[S] (*|name[,...])
TO grantee
```

Graph Privileges



TRAVERSE

READ
MATCH

MERGE
SET PROPERTY

CREATE
DELETE

{*|*property-name*[,...]}

ON GRAPH[S] *|*name*[,...]

ELEMENT[S]
NODES[S]
RELATIONSHIP[S]

*|*name*[,...]

TO *grantee*

ALL [[GRAPH] PRIVILEGES]
WRITE

SET LABEL
REMOVE LABEL

*|*label-name*[,...]

ON GRAPH[S] *|*name*[,...]

SHOW PRIVILEGES AS COMMANDS

What is it

- New option for SHOW PRIVILEGES
  - return the CYPHER commands to create the privileges
  - or the CYPHER commands to revoke the privileges
  - Added to the System, Role and User versions of the SHOW command

- Syntax:

```
SHOW PRIVILEGES [AS [REVOKE] COMMAND[S]]
SHOW ROLE role PRIVILEGES [AS [REVOKE] COMMAND[S]]
SHOW USER user PRIVILEGES [AS [REVOKE] COMMAND[S]]
```

What is it for

For DBAs to easily retrieve the CYPHER commands to create or revoke existing privileges.

Show privileges commands - examples

```
SHOW ROLE admin PRIVILEGES AS COMMANDS
-------------------------------------
"GRANT ACCESS ON DATABASE * TO `admin`"
"GRANT ALL DBMS PRIVILEGES ON DBMS TO `admin`"
"GRANT CONSTRAINT MANAGEMENT ON DATABASE * TO `admin`"
"GRANT INDEX MANAGEMENT ON DATABASE * TO `admin`"
"GRANT MATCH {*} ON GRAPH * NODE * TO `admin`"
"GRANT MATCH {*} ON GRAPH * RELATIONSHIP * TO `admin`"
"GRANT NAME MANAGEMENT ON DATABASE * TO `admin`"
"GRANT START ON DATABASE * TO `admin`"
"GRANT STOP ON DATABASE * TO `admin`"
"GRANT TRANSACTION MANAGEMENT (*) ON DATABASE * TO
`admin`"
"GRANT WRITE ON GRAPH * TO `admin`"
```

For users, roles are parameterized. The idea is that the output could be used to create a new role, based on the privileges of a user.

```
SHOW USER jake PRIVILEGES AS COMMAND
------------------------------------
"GRANT ACCESS ON DATABASE `neo4j` TO $role"
"GRANT ACCESS ON DEFAULT DATABASE TO $role"
"GRANT EXECUTE PROCEDURE * ON DBMS TO $role"
```

Output the revoke version of a grant/deny.

```
SHOW ROLE architect PRIVILEGES AS REVOKE COMMANDS WHERE command CONTAINS 'MATCH'
--------------------------------------------------------------------------------
"REVOKE GRANT MATCH {*} ON GRAPH * NODE * FROM `architect`"
"REVOKE GRANT MATCH {*} ON GRAPH * RELATIONSHIP * FROM `architect`"
```

Backup/Restore

Backup / Restore for Multi-Database

What is it?

- Wildcards for database names (allowing multiple backups in one command)
- Users, roles and permissions are optionally extracted to the backup.

What is it for?

- Multi-tenancy scenarios
  - I sell services from a shared infrastructure to multiple customers
- Re-architecting
  - I need to organise differently; for scale, performance, compliance…
- Migrations
  - I'm moving databases between installations of Neo4j. DC to DC, Production to UAT, redundant DR etc.

Backup / Restore Syntax

```
USAGE
neo4j-admin backup [--check-consistency] [--fallback-to-full] [--verbose]
                   [--additional-config=<path>] --backup-dir=<path>
                   [--check-graph=<true/false>]
                   [--check-index-structure=<true/false>]
                   [--check-indexes=<true/false>]
                   [--check-label-scan-store=<true/false>]
                   [--check-property-owners=<true/false>]
                   [--check-relationship-type-scan-store=<true/false>]
                   [--database=<database>] [--from=<host:port>]
                   [--include-metadata=<all/users/roles>]
[--pagecache=<size>]
                   [--report-dir=<path>]

OPTIONS

     ...
     --database=<database> Name of the remote database to backup. Can
contain
                             * and ? for globbing.
                             Default: neo4j
     ...
     --include-metadata=<all/users/roles>
                           Include metadata in file. Can't be used for
backing
                             system database.
                           roles - commands to create the roles and
privileges
                             (for both database and graph) that affect the
use
                             of the database
                           users - commands to create the users that can
use
                             the database and their role assignments
                           all - include roles and users
```

```
USAGE

neo4j-admin restore [--force] [--move] [--verbose] [--database=<database>]
                    [--to-data-directory=<path>]
                    [--to-data-tx-directory=<path>]
                     --from=<path>[,<path>...]...

OPTIONS

      --verbose   Enable verbose output.
      --from=<path>[,<path>...]...
                  Path or paths from which to restore. Every path can contain
                    asterisks or question marks in the last subpath. Multiple
                    paths may be separated by a comma, but paths themselves
                    must not contain commas.
      --database=<database>
                  Name of the database after restore. Usage of this option is
                    only allowed if --from parameter point to exact one
                    directory
      --force     If an existing database should be replaced.
      --move      Moves the backup files to the destination, rather than
                    copying.
      --to-data-directory=<path>
                  Base directory for databases. Usage of this option is only
                    allowed if --from parameter point to exact one directory
      --to-data-tx-directory=<path>
                  Base directory for transaction logs. Usage of this option
is
                    only allowed if --from parameter point to exact one
                    directory
```

neo4j-admin copy - sharding a database (4.2)

What is it

Additional options for neo4j-admin copy:

```
--keep-only-nodes-with-labels
--skip-nodes-with-labels
--skip-node-properties
--skip-relationship-properties
--keep-only-node-properties
--keep-only-relationship-properties
```

Example:

```
$neo4j-home> bin/neo4j-admin copy
                          --from-database=neo4j \
                          --to-database=shard1 \
                          --keep-only-nodes-with-labels=S1,SAll
```

What is it for

Fabric supports partitioning of data across databases.

Users can split an existing database after it has grown in size, with all its implications.

The additional options can be used to filter out data and migrate to a Fabric installation

CREATE DATABASE with seed

What it is

- Additional Neo4j 4.3 **CREATE DATABASE** options to copy, restore or load into a standalone or cluster DBMS.
- **CREATE DATABASE** command can include a map of OPTIONS

```
CREATE DATABASE name [IF NOT EXISTS] OPTIONS
   { existingData: data-option,
     existingDataSeedInstance: 'instance-id'}
```

data-option:

```
'use'       : use existing data (this is the only
              option in Neo4j 4.3), throw error if
              no data is present
```

What it is for

For DBAs who want to restore a database archive to a single cluster instance, without needing to perform restores on the remaining instances.

For DBAs who want to maintain a single script, for standalone or clusters, to create databases using archived data.

Backup/Restore Tutorial

Tutorial

Back up and restore a database in Causal Cluster Enterprise Edition

> *This tutorial provides a detailed example of how to back up and restore a database in a running Causal Cluster.*

The following example assumes that you want to restore a database backup, which has users and roles associated with it, in a running Causal Cluster with three core servers. For more information on how to set up a Causal Cluster with three cores, see Set up a local Causal Cluster.

> In a Neo4j DBMS, every database is backed up individually. Therefore, it is very important to plan your backup strategy for each of them. For more detailed information on how to design an appropriate backup strategy for your setup, see Backup and restore.

neo4j

Monitoring

Metrics

Global Metrics

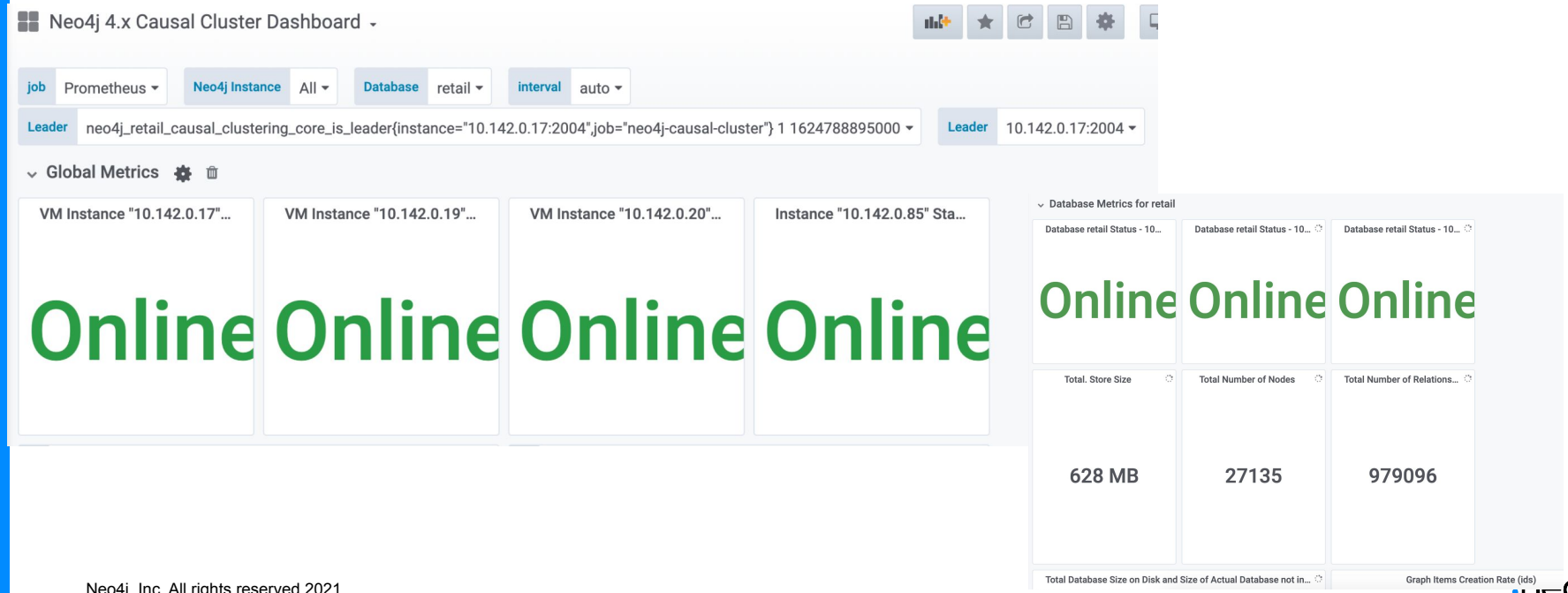
- Global metrics cover the whole database management system, and represents the status of the system as a whole
    - Page cache metrics
    - GC metrics
    - Thread Metrics
    - Memory pool metrics
    - Memory buffers metrics
    - File descriptor metrics
    - Database operation metrics
    - Bolt metrics
    - Web Server metrics

Database Metrics

- Metrics specific to each individual database
    - Transaction metrics
    - Checkpoint metrics
    - Log rotation metrics
    - Database data metrics
    - Cypher metrics
    - Causal clustering metrics

Metrics

[Best Practices for Monitoring Neo4j Enterprise with Prometheus and Grafana]



Leveraging Causal Clustering Server Groups

Cluster Leadership Control and Balancing

What is it

- Leadership Transfer Extension: extension to raft protocol that allows servers to transfer leadership
- Control leadership based on user priorities
- In multiple databases scenario, balance leadership equally or based on user priorities

What is it for

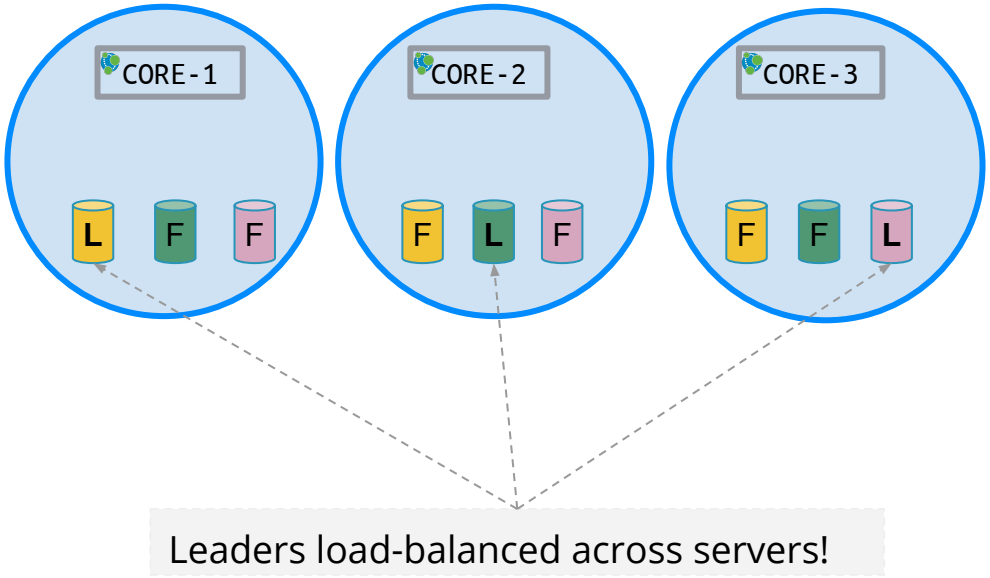
- Sometimes a leader must step down (e.g. maintenance). The leader will now avoid a new election delay by passing leadership before stepping down.
- Some servers might be more suitable to become leaders than others. That preference can now be configured.
- In clusters with several (possibly hundreds) of databases, it is now possible to load balance leadership load.

Scenario 1: Load Balancing Leadership

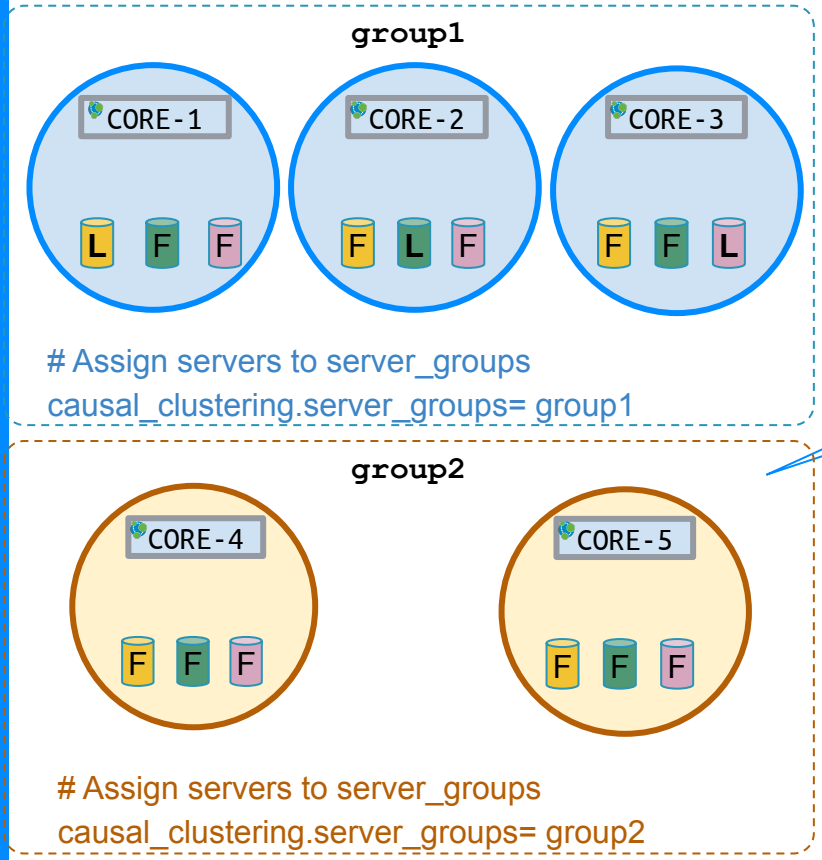
```
causal_clustering.leadership_balancing=equal_balancing
# do NOT define priority groups for the databases
```

a. The load-balancer algorithm runs periodically and if needed requests transfer of leaderships.

b. When a leader steps down, leadership is passed to any suitable member, avoiding election
   i. Load-balancer can request transfer



Scenario 2: Preferred Leaders




```
# declare r1 as the priority group for all databases
causal_clustering.leadership_priority_group.dbYellow=group1
causal_clustering.leadership_priority_group.dbGreen=group1
causal_clustering.leadership_priority_group.dbPink=group1
```

No Leaders!

- Over time, the leader for the database will end up being a member of group1
  - Checks run periodically
  - Leadership transfer triggered when required.
- When a Leader steps down, it passes leadership to a member of the priority group

Cluster Testing with 500 Databases

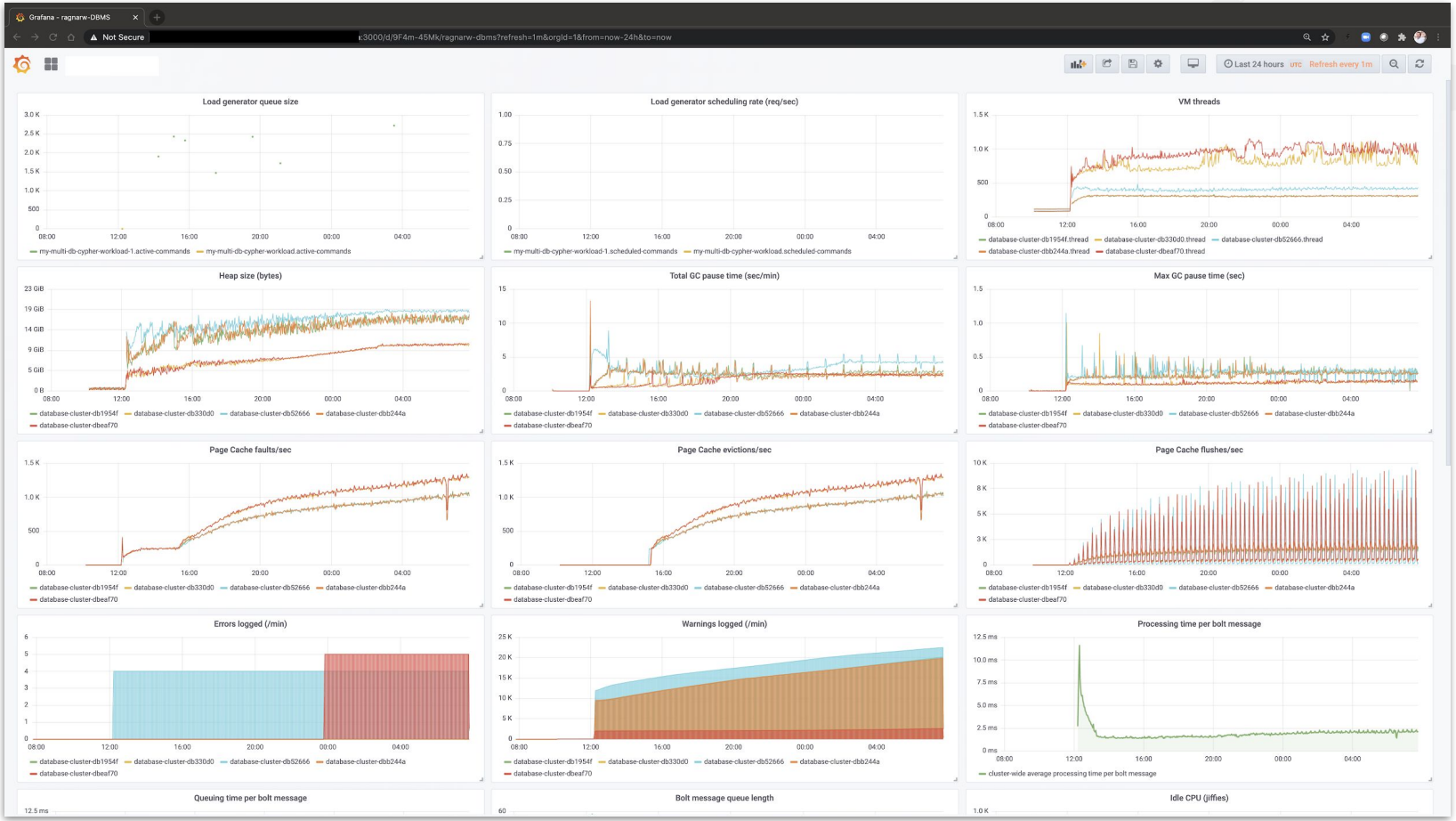
What is it

- Phase 2 of our effort in supporting multiple databases with Cluster.
- In 4.1, we tested the execution of common user and admin operations with 200 concurrent databases.
- In 4.2, we pushed the testing to 500 concurrent databases.
- We are confident Neo4j can support more than 500 concurrent databases, but our testing efforts (for now) is limited to this number.
- Things to note in the charts:
  - Stabilisation once the databases have been created
  - Load balancing on the members of the cluster

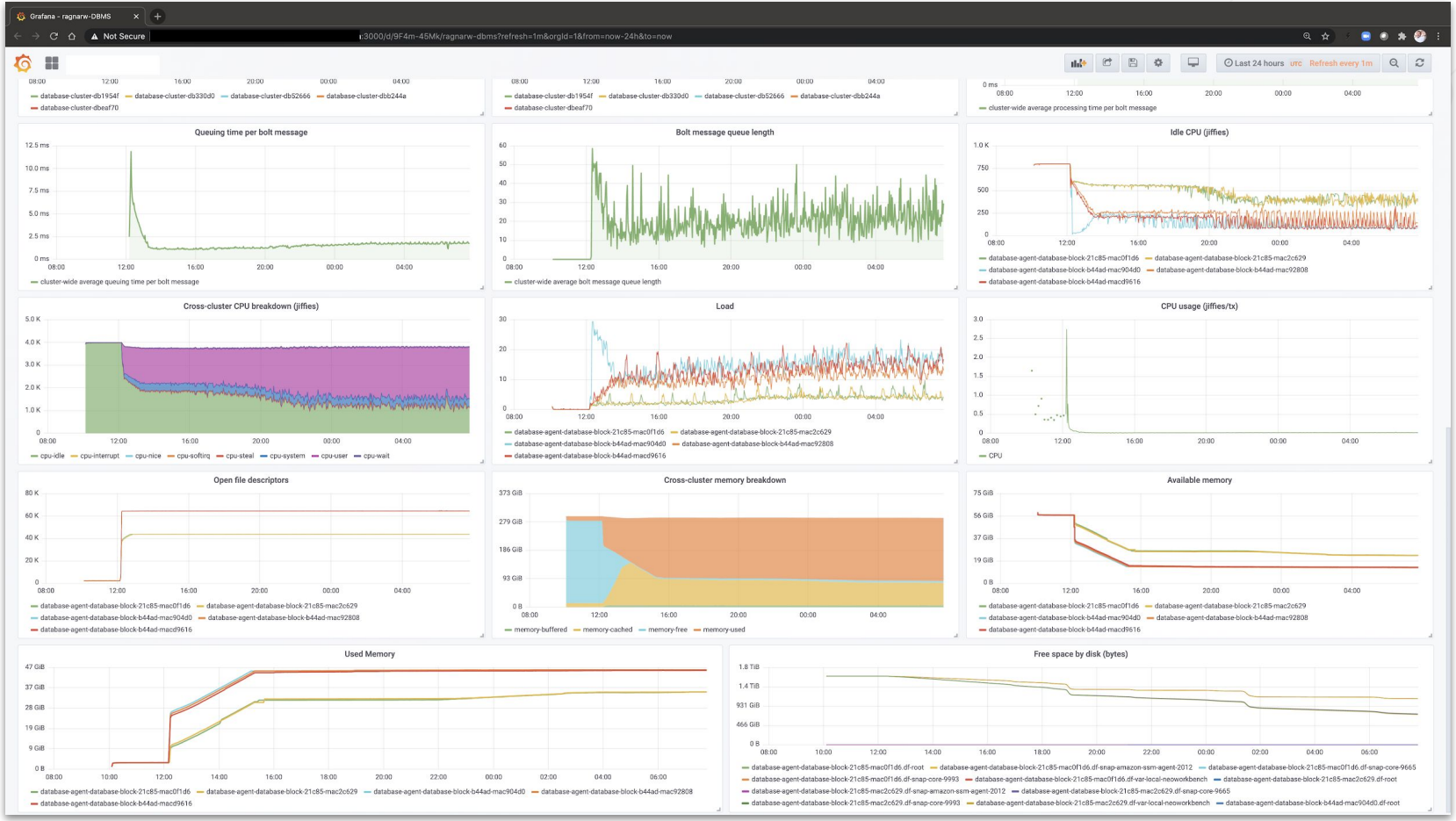
What is it for

- We expanded our testing efforts to provide extra safety for SaaS providers and customers who require a large number of databases.

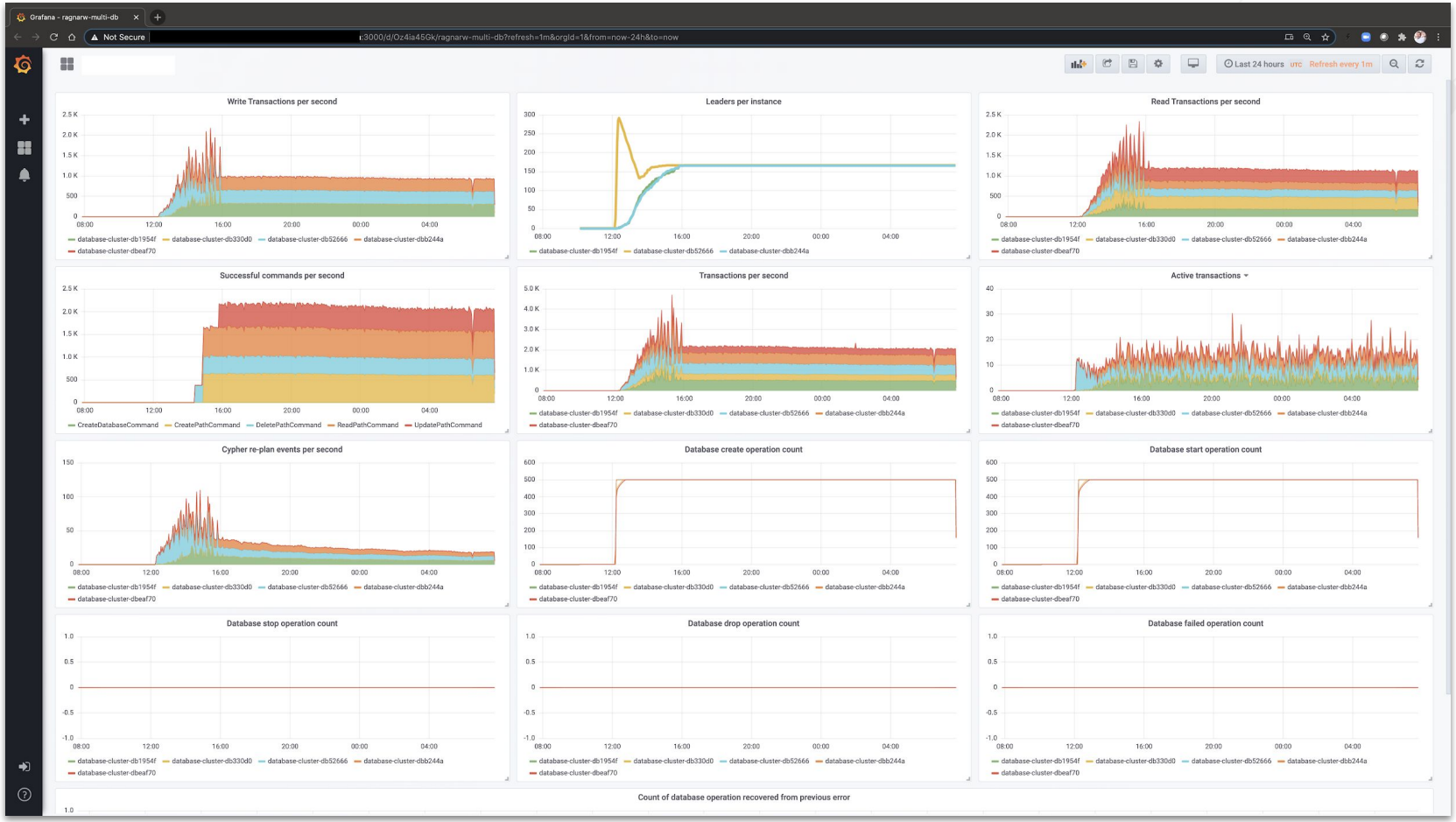
Cluster Testing with 500 Databases



Cluster Testing with 500 Databases



Cluster Testing with 500 Databases



Onboarding a New Database

Bring it all together

- Define a process for onboarding/provisioning a New Database
  - Create / Seed the database
  - Ensure there’s enough space
  - Add / assign roles and users
  - Add to backup process
  - Ensure properly Monitored
  - Add to Server Group if necessary

WAIT/NOWAIT in Database management

What is it

- In 4.0 and 4.1, when a database is created or dropped, the command is passed to the server and it is executed in background: the user receives an acknowledge that the command has been received, not that it has been executed.
- In 4.2, the administrator may decide to use the default behaviour (as in 4.0/4.1) or to wait for a given time in seconds, or until the command has been completed, or an error occurred.
- It works with CREATE|DROP|START|STOP DATABASE
- In future versions, WAIT and NOWAIT will be available in most of the Cypher admin and DML commands.

```
{CREATE|DROP|START|STOP} DATABASE name
                         [NOWAIT|WAIT [n [SECOND[S]]]]
```

What is it for

- In Cluster, CREATE DATABASE and DROP DATABASE are operations that may take seconds, and during that period the user cannot operate on that database (or for example recreate it if it has been dropped). The WAIT command can be used in scripts or applications to avoid to poll the server to verify if the command has been executed.

Sample RBAC Cypher

```
//Create Roles
create role read_restricted;
create role readwrite_restricted;
//Creating Users
CREATE USER rr_user SET PASSWORD "letmein" CHANGE NOT REQUIRED;
CREATE USER rw_user SET PASSWORD "letmein" CHANGE NOT REQUIRED;
//Grant role to user
grant role read_restricted to rr_user;
grant role readwrite_restricted to rw_user;
//Assigning Privileges
GRANT ACCESS      ON DATABASE fraud          TO read_restricted;
GRANT MATCH       {*}   ON GRAPH fraud          TO
read_restricted;
DENY  read {ssn}  ON GRAPH fraud NODES SSN TO read_restricted;
//Access SSN node and you will not be able to read any property
with name ssn
MATCH (s:SSN) return s limit 10;
//Now for rw_user
//Assign privileges to rw_user
GRANT ACCESS      ON DATABASE fraud          TO
readwrite_restricted;
GRANT MATCH       {*}   ON GRAPH fraud          TO
readwrite_restricted;
GRANT SET PROPERTY {*}   ON GRAPH fraud NODE *      TO
readwrite_restricted;
GRANT CREATE ON GRAPH fraud ELEMENTS * TO readwrite_restricted;
DENY  SET PROPERTY {ssn}  ON GRAPH fraud NODES SSN TO
readwrite_restricted;
```

```
/So if I try to create a node with property ssn it fails
create(s:SSN {ssn:"123-45-678"}) return s;
//So if I try to create a node with property id it succeeds.
create(s:SSN {id:"123-45-678"}) return s
////////////////////////////////
//DB Admin
//Create Role DBAdmin
create role dbadmin;
//create user dba
CREATE USER dba SET PASSWORD "letmein" CHANGE NOT REQUIRED;
grant role dbadmin to dba;
//DBA Admin privileges but does not have read access on DB
GRANT ACCESS      ON DATABASE fraud          TO dbadmin;
GRANT INDEX       MANAGEMENT ON DATABASE fraud TO dbadmin;
GRANT CONSTRAINT  MANAGEMENT ON DATABASE fraud TO dbadmin;
GRANT TRANSACTION MANAGEMENT ON DATABASE fraud TO dbadmin;
GRANT USER        MANAGEMENT ON DBMS TO dbadmin;
GRANT DATABASE MANAGEMENT on DBMS * to dbadmin
//create index
create index on :Client(name);
//drop users and roles
drop role dbadmin;
drop user dba;
drop role read_restricted;
drop role readwrite_restricted;
drop user rr_user;
drop user rwr_user;
```

```
////////////////////////////
//after setting this value
CALL
dbms.setConfigValue("dbms.memory.transaction.max_siz
e","2MiB")
CALL
dbms.setConfigValue("dbms.memory.transaction.databab
ase_max_size","10MiB")
CALL
dbms.setConfigValue("dbms.memory.transaction.global_
max_size","10MiB")
CALL
dbms.setConfigValue("dbms.memory.transaction.max_siz
e","0B");
CALL
dbms.setConfigValue("dbms.memory.transaction.databab
ase_max_size","0B");
CALL
dbms.setConfigValue("dbms.memory.transaction.global_
max_size","0B");
//run the query
match (c:Client)-[:PERFORMED]->(n:Transaction)
return c.name, sum(n.amount) as Total_Amount order
by Total_Amount
// Example of User Managerment - ISIM Manager
CREATE ROLE isimManager;
GRANT USER MANAGEMENT ON DBMS TO isimManager;
GRANT ROLE MANAGEMENT ON DBMS TO isimManager;
GRANT SHOW PRIVILEGE ON DBMS TO isimManager;
GRANT ASSIGN PRIVILEGE ON DBMS TO isimManager
GRANT ASSIGN ROLE ON DBMS TO isimManager;
create User isimUser;
CREATE USER isimUser SET PASSWORD "password" CHANGE
NOT REQUIRED;
grant role isimManager to isimUser;
CREATE USER alice SET PASSWORD "password" CHANGE NOT
REQUIRED;
grant role architect to alice;
CREATE ROLE isimManager2 IF NOT EXISTS AS COPY OF
isimManager, reader;
//Revoking privileges from PUBLIC role.
REVOKE ACCESS ON DEFAULT DATABASE from PUBLIC
REVOKE EXECUTE PROCEDURE * ON DBMS FROM PUBLIC
REVOKE EXECUTE FUNCTIONS * ON DBMS FROM PUBLIC
```

LDAP - dynamic settings & prevent inadvertent authorisation

What it is

1. Dynamic LDAP settings;
   `dbms.security.ldap.authentication.attribute`
   `dbms.security.ldap.authorization.user_search_filter`
   `dbms.security.ldap.authorization.user_search_base`
   `dbms.security.ldap.authentication.user_dn_template`
   `dbms.security.ldap.authorization.group_membership_attributes`

2. Setting to restrict LDAP authorisation to a particular group;
   `dbms.security.ldap.authorization.access_permitted_group`

3. Custom attribute for looking up LDAP users (avail. in the plugin)
   `dbms.security.ldap.authentication.attribute`
   Enabled with a switch
   `dbms.security.ldap.authentication.search_for_attribute`
   This new setting deprecates, and replaces
   `dbms.security.ldap.authentication.use_samaccountname`

What it is for

LDAP/Active Directory integration can be a difficult and error-prone pursuit of the correct configuration.

Some customers update settings such as group-to-role mappings frequently as part of a provisioning workflow.

A new setting will restrict the existing authorisation settings prevent inadvertently grant ing all members of a directory PUBLIC access to the Neo4j DBMS.

A plugin is no longer required for customers who need to map to a custom attribute in the directory.