



Monitoring Neo4J

Create some nodes




- `cypher-shell -u neo4j -p secret -d neo4j --format plain`
 - `:use neo4j;`
 - `MATCH (n) DETACH DELETE n;`
 - `#Create some records`
 - `FOREACH (i IN RANGE(1,200) | CREATE (:Person {name:'Person' + i}));`

Exercise: Monitoring queries

- `cypher-shell -u neo4j -p secret -d neo4j --format plain`
 - #Execute a query that runs for longer than 1000 ms:
 - `MATCH (a), (b), (c), (d) RETURN count(id(a));`





Viewing currently running queries



\$




To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

\$:queries



Database URI	User	Query	Params	Meta	Elapsed time	Kill
bolt://ec2-3-82-2-121.compu...	admin	CALL dbms.listQueries	{}	{}	0 ms	
bolt://ec2-3-82-2-121.compu...	reader	MATCH (a), (b), (c), (d), (e) RETU...	{}	{ "type": "user-direct", "app": "n...	1344899 ms	

Found 2 queries running on one server

AUTO-REFRESH ☐ OFF 



Killing a long-running query

\$

To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

X

\$:queries

Database URI	User	Query	Params	Meta	Elapsed time	Kill
bolt://ec2-3-82-2-121.compu...	admin	CALL dbms.listQueries	{}	{}	0 ms	
bolt://ec2-3-82-2-121.compu...	reader	MATCH (a), (b), (c), (d), (e) RETU...	{}	{ "type": "user-direct", "app": "n...	1344899 ms	

Found 2 queries running on one server

AUTO-REFRESH ☐ OFF



Automating monitoring of queries

- Automate the killing of long-running queries:
 - `CALL dbms.listQueries() YIELD query, elapsedTimeMillis, queryId, username`
 - `WHERE NOT query CONTAINS toLower('LOAD')`
 - `AND elapsedTimeMillis > 1000`
 - `WITH query, collect(queryId) AS q`
 - `CALL dbms.killQueries(q) YIELD queryId`
 - `RETURN query, queryId;`

Monitoring connections

- Can view the current connections to a Neo4j instance from cypher-shell
 - Call `dbms.listConnections()`;

```
[neo4j> call dbms.listConnections();
+-----+
| connectionId | connectTime           | connector | username | userAgent                                     | serverAddress | clientAddress |
+-----+
| "bolt-97"    | "2019-02-19T20:35:30.513Z" | "bolt"    | "admin"   | "neo4j-javascript/1.7.2"                   | "172.31.24.255:7687" | "216.45.71.93:57415" |
| "bolt-840"   | "2019-02-19T21:58:04.65Z"  | "bolt"    | "publisher" | "neo4j-java/1.7.2-7165cef0d4b602da30b65613977744ad661c2e67" | "127.0.0.1:7687" | "127.0.0.1:40344" |
| "bolt-779"   | "2019-02-19T21:52:05.978Z" | "bolt"    | "admin"   | "neo4j-java/dev"                           | "127.0.0.1:7687" | "127.0.0.1:40338" |
| "bolt-839"   | "2019-02-19T21:58:04.358Z" | "bolt"    | "publisher" | "neo4j-java/1.7.2-7165cef0d4b602da30b65613977744ad661c2e67" | "127.0.0.1:7687" | "127.0.0.1:40340" |
+-----+

4 rows available after 1 ms, consumed after another 0 ms
neo4j> |
```

- Terminate the connection
 - `dbms.killConnection()`

```
[neo4j> CALL dbms.killConnections(['bolt-1362','bolt-1363']);
+-----+
| connectionId | username   | message           |
+-----+
| "bolt-1362"  | "publisher" | "Connection found" |
| "bolt-1363"  | "publisher" | "Connection found" |
+-----+

2 rows available after 14 ms, consumed after another 0 ms
neo4j> |
```

A composite image featuring a central handshake. The hands are positioned in the center, with one hand appearing slightly more prominent. The background is a blurred cityscape with tall buildings. A large, semi-transparent teal shape, resembling a stylized 'A' or a large arrow, is overlaid on the image. In the upper left, there's a dark, semi-transparent rectangle containing a faint bar chart. On the right side, there's a circular inset showing a cityscape with a line graph and the text 'Average 45%'. The overall color palette is dominated by blues, greys, and the teal of the overlay.

THANK YOU