Monitoring

neo4j

What are we going to do?

- View currently running queries
- Kill running queries
- Setup metrics

Killing long running queries

Currently running queries

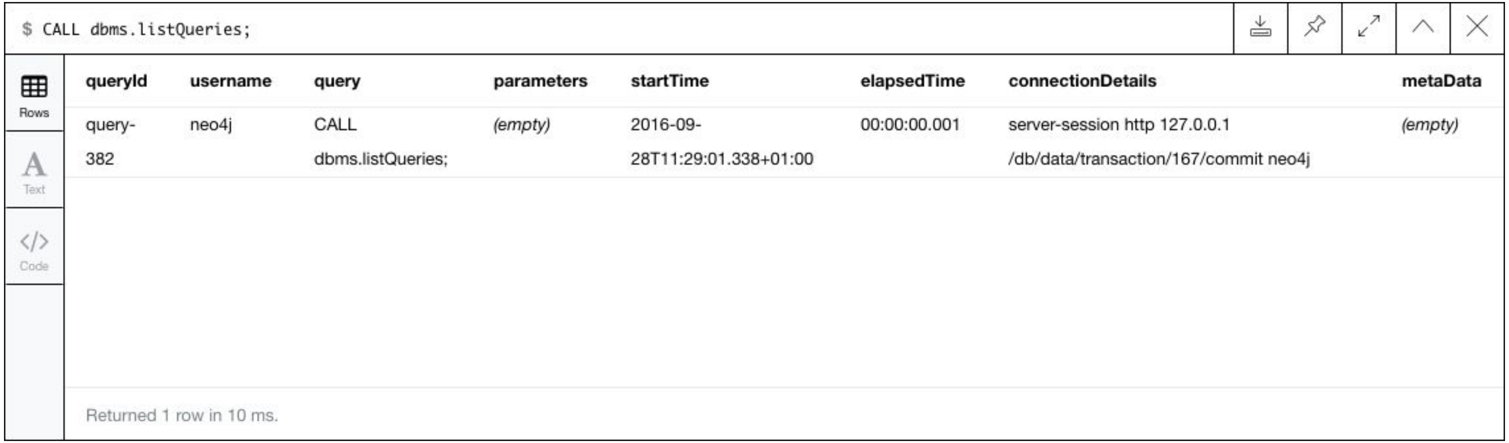
We can run the following procedure to view the currently running queries:

```
CALL dbms.listQueries();
```

Currently running queries

We can run the following procedure to view the currently running queries:

```
CALL dbms.listQueries();
```



```
$ CALL dbms.listQueries;
```

queryId	username	query	parameters	startTime	elapsedTime	connectionDetails	metaData
query-382	neo4j	CALL dbms.listQueries;	*(empty)*	2016-09-28T11:29:01.338+01:00	00:00:00.001	server-session http 127.0.0.1 /db/data/transaction/167/commit neo4j	*(empty)*

Returned 1 row in 10 ms.

Start a long running query

Let’s create a long running query and then kill it. Execute the following query:

```
MATCH (m), (n), (o), (p), (q)
RETURN COUNT(*) AS count
```

Currently running queries

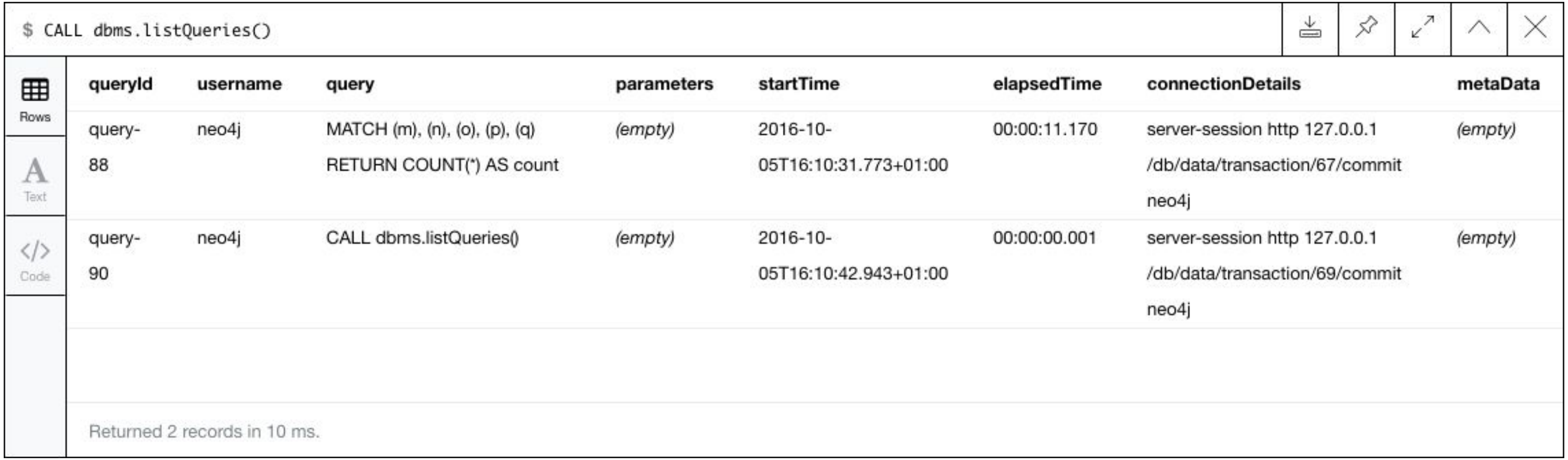
Now run the list queries procedure again:

```
CALL dbms.listQueries()
```

Currently running queries

Now run the list queries procedure again:

```
CALL dbms.listQueries()
```



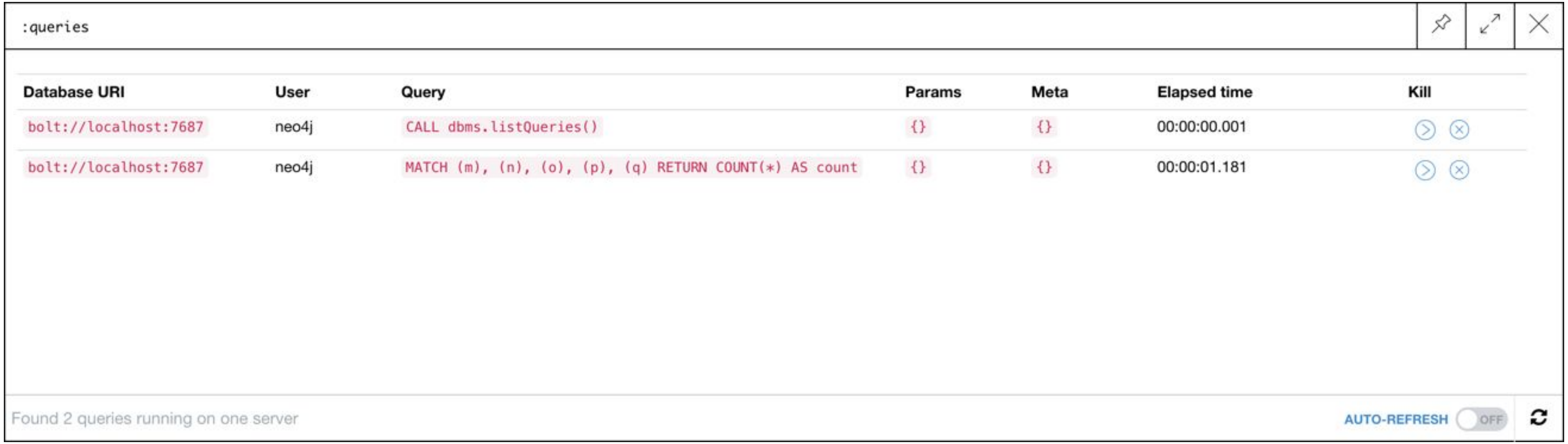
```
$ CALL dbms.listQueries()
```

queryId	username	query	parameters	startTime	elapsedTime	connectionDetails	metaData
query-88	neo4j	MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count	*(empty)*	2016-10-05T16:10:31.773+01:00	00:00:11.170	server-session http 127.0.0.1 /db/data/transaction/67/commit neo4j	*(empty)*
query-90	neo4j	CALL dbms.listQueries()	*(empty)*	2016-10-05T16:10:42.943+01:00	00:00:00.001	server-session http 127.0.0.1 /db/data/transaction/69/commit neo4j	*(empty)*

Returned 2 records in 10 ms.

Currently running queries

We have a built in tool in the browser. Type `:queries` into the query pane:



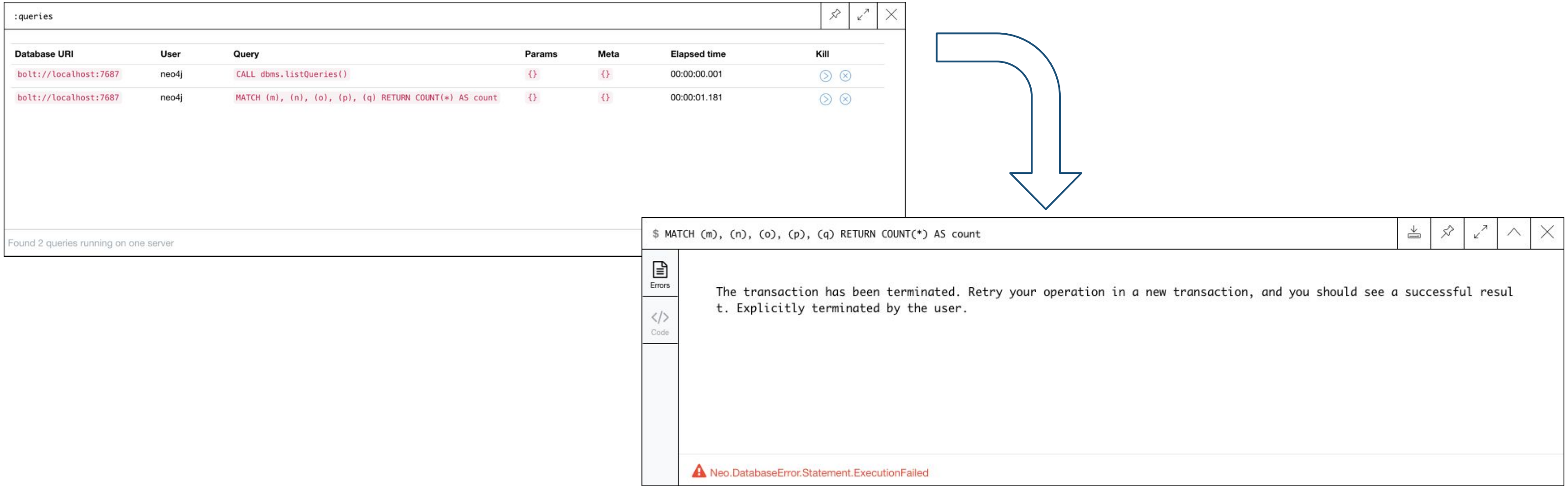
```
:queries
```

Database URI	User	Query	Params	Meta	Elapsed time	Kill
`bolt://localhost:7687`	neo4j	`CALL dbms.listQueries()`	{}	{}	00:00:00.001	⊳ ⊗
`bolt://localhost:7687`	neo4j	`MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count`	{}	{}	00:00:01.181	⊳ ⊗

Found 2 queries running on one server

Killing a query

We have a built in tool in the browser. Type `:queries` into the query pane:



:queries

Database URI	User	Query	Params	Meta	Elapsed time	Kill
bolt://localhost:7687	neo4j	`CALL dbms.listQueries()`	{}	{}	00:00:00.001	⊙ ⊗
bolt://localhost:7687	neo4j	`MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count`	{}	{}	00:00:01.181	⊙ ⊗

Found 2 queries running on one server

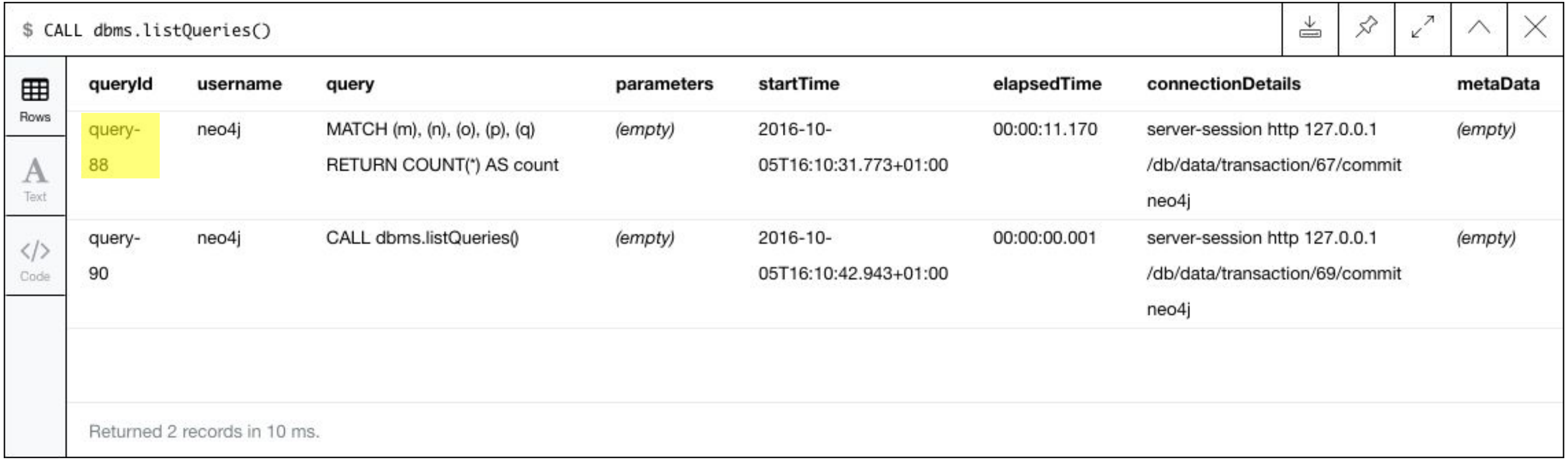
```
$ MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count
```

```
The transaction has been terminated. Retry your operation in a new transaction, and you should see a successful resul
t. Explicitly terminated by the user.
```

⚠ Neo.DatabaseError.Statement.ExecutionFailed

Kill running queries

We can also kill the query manually using the kill query procedure:



```
$ CALL dbms.listQueries()
```

queryId	username	query	parameters	startTime	elapsedTime	connectionDetails	metaData
query-88	neo4j	MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count	*(empty)*	2016-10-05T16:10:31.773+01:00	00:00:11.170	server-session http 127.0.0.1 /db/data/transaction/67/commit neo4j	*(empty)*
query-90	neo4j	CALL dbms.listQueries()	*(empty)*	2016-10-05T16:10:42.943+01:00	00:00:00.001	server-session http 127.0.0.1 /db/data/transaction/69/commit neo4j	*(empty)*

Returned 2 records in 10 ms.

```
CALL dbms.killQuery("query-88")
```

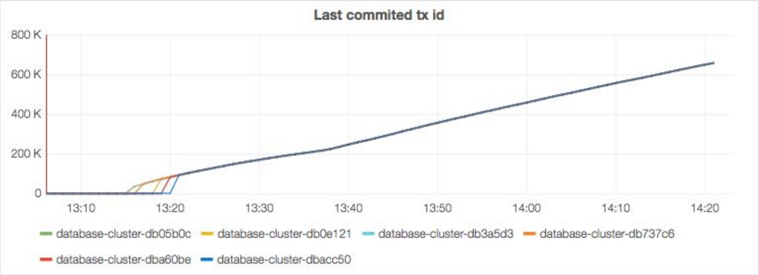
Metrics

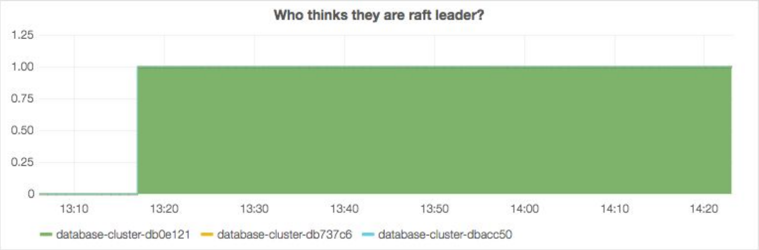
neo4j

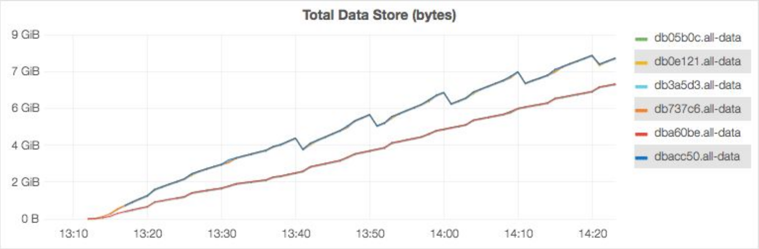
Enabling metrics

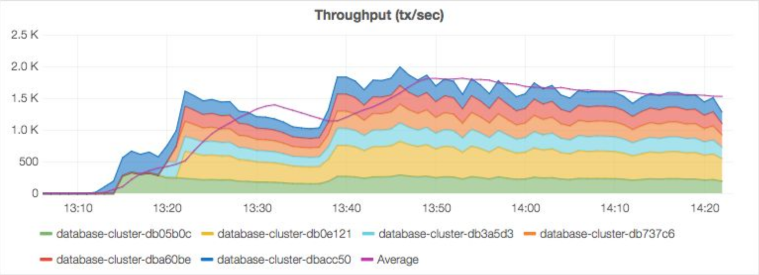
```
// default setting for enabling all supported metrics

metrics.enabled=true

metrics.graphite.enabled=true

metrics.graphite.server=<ip>:2003

metrics.graphite.interval=<how often to send data, defaults to 3s>

metrics.prefix=<Neo4j instance name, e.g. wwwneo1>
```

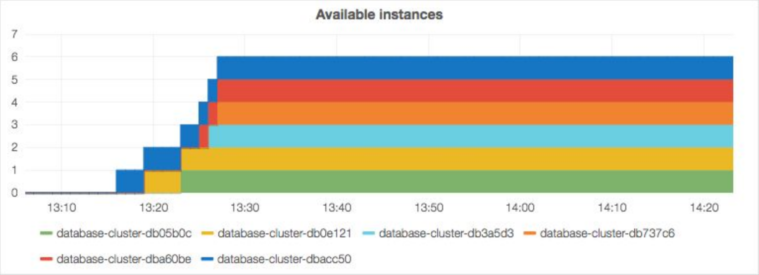
Grafana

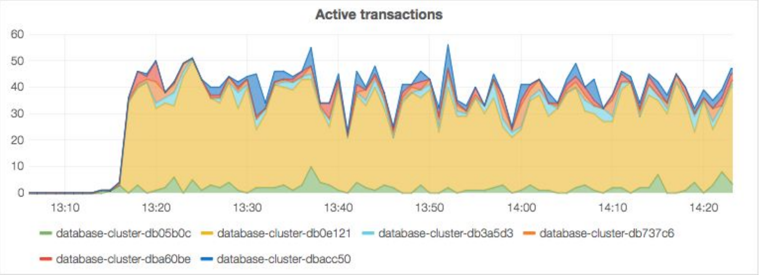

Last commited tx id


Who thinks they are raft leader?


Total Data Store (bytes)


Throughput (tx/sec)


Available instances


Active transactions

End of Module Monitoring

Questions ?