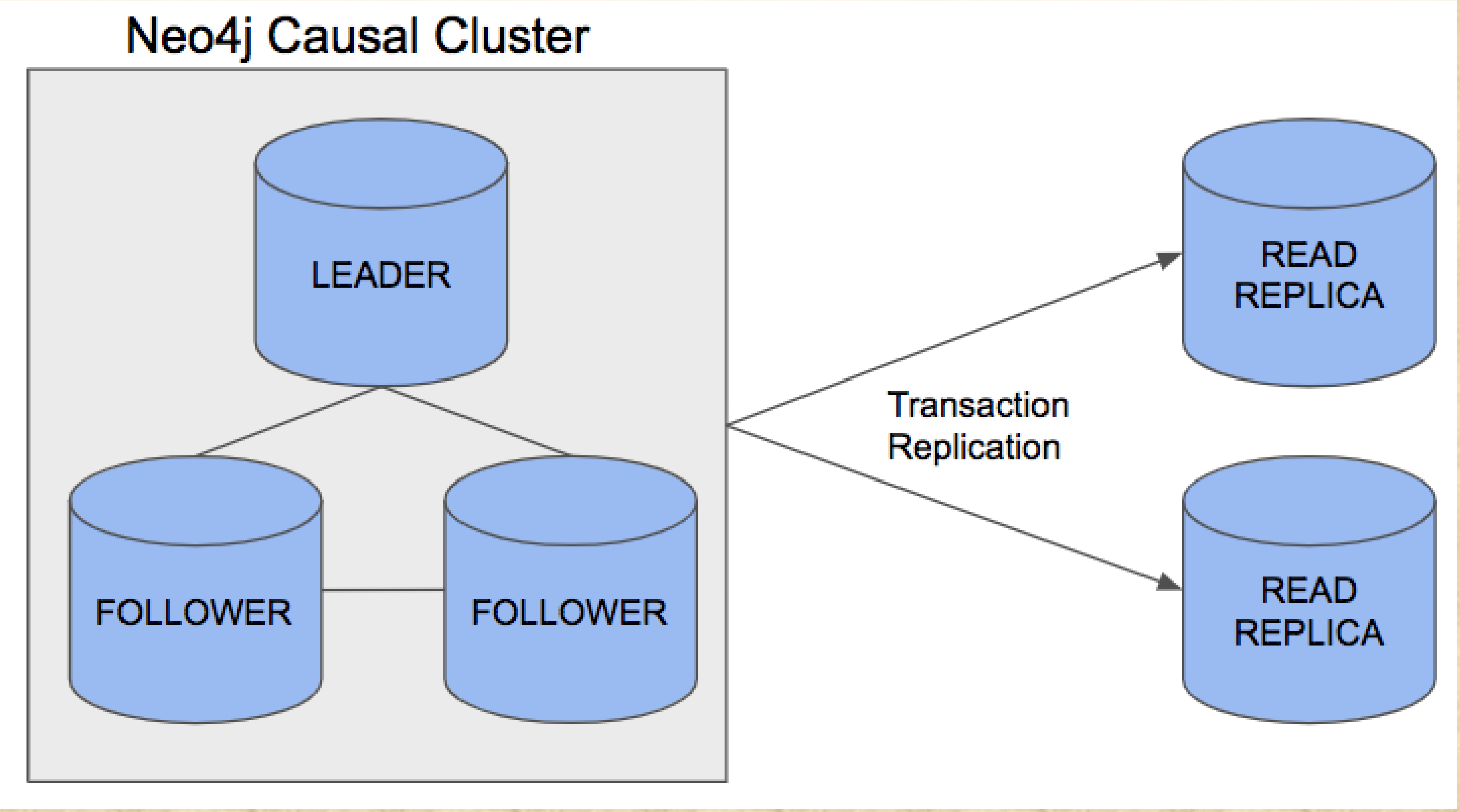
Neo4j

Causal Clustering

Neo4j Causal Clustering

- A cluster is composed of three or more Neo4j instances that communicate with one another to provide fault-tolerance and high-availability
- Uses a **consensus protocol (RAFT)** to coordinate the cluster
- each database has a perfect, complete copy of the entire database (no partitioning)
- Each machine in the cluster has a “**role**” –
  - **Leader or**
  - **Follower**

Cluster Architecture



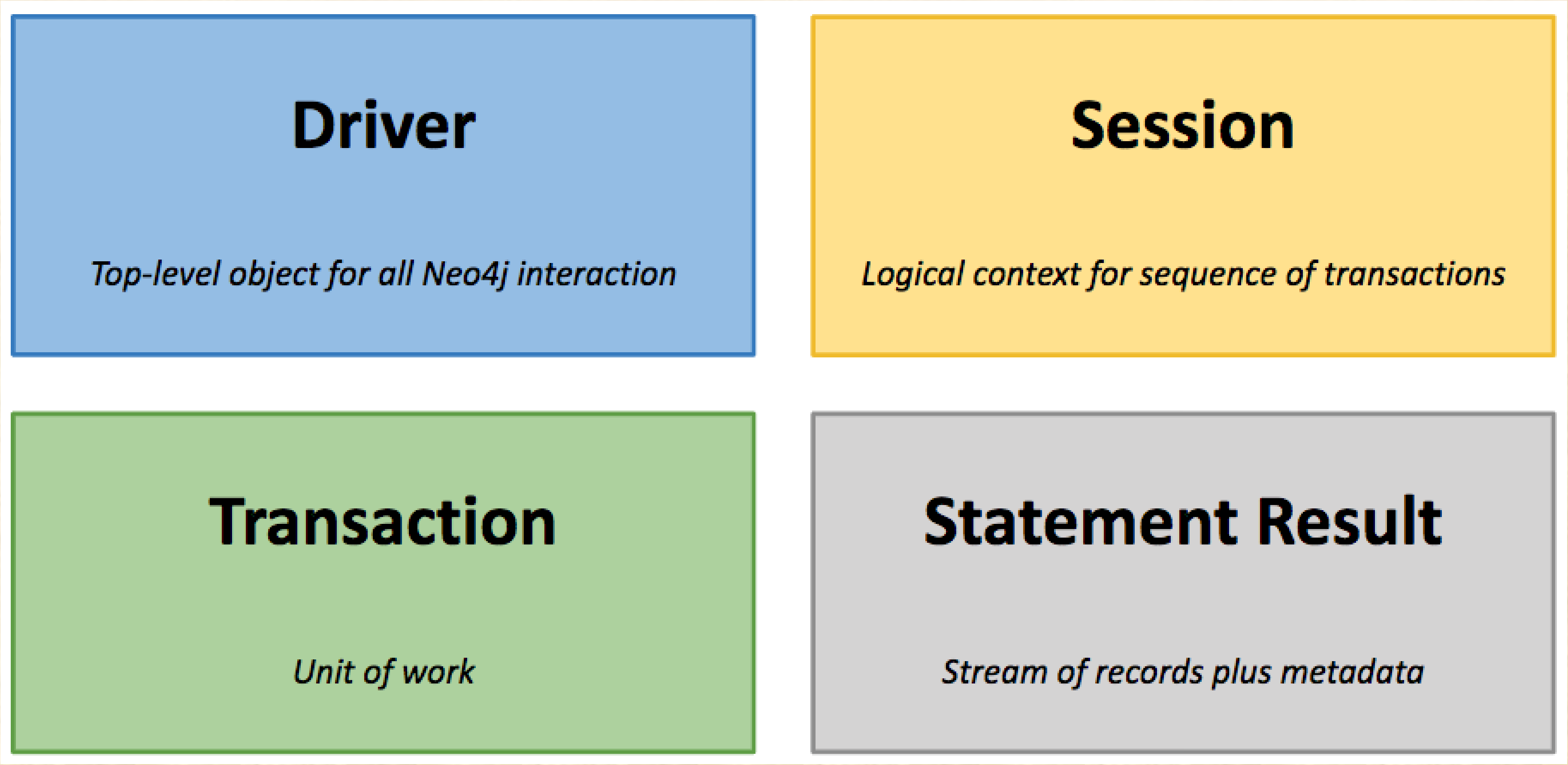
Cluster Roles

- The **leader** is responsible for coordinating the cluster and accepting all writes
- **Followers** help scale the read workload ability of the cluster and provide for high-availability of data
- If any one follower fails, show continue
- can have any number caches in the form of read replicas

Topology changes

- In the lifecycle of a cluster, cluster roles are temporary.
- Suppose you have machines A, B, and C.
- If A fails, then the remaining nodes (B and C) will elect a new leader amongst themselves.
- When A restarts, later on, it will rejoin the cluster, but probably as a follower.
- Roles can change through the lifecycle of the cluster
- Role changes are not a cause for concern

Driver API consists of 4 key parts



Driver

Top-level object for all Neo4j interaction

Session

Logical context for sequence of transactions

Transaction

Unit of work

Statement Result

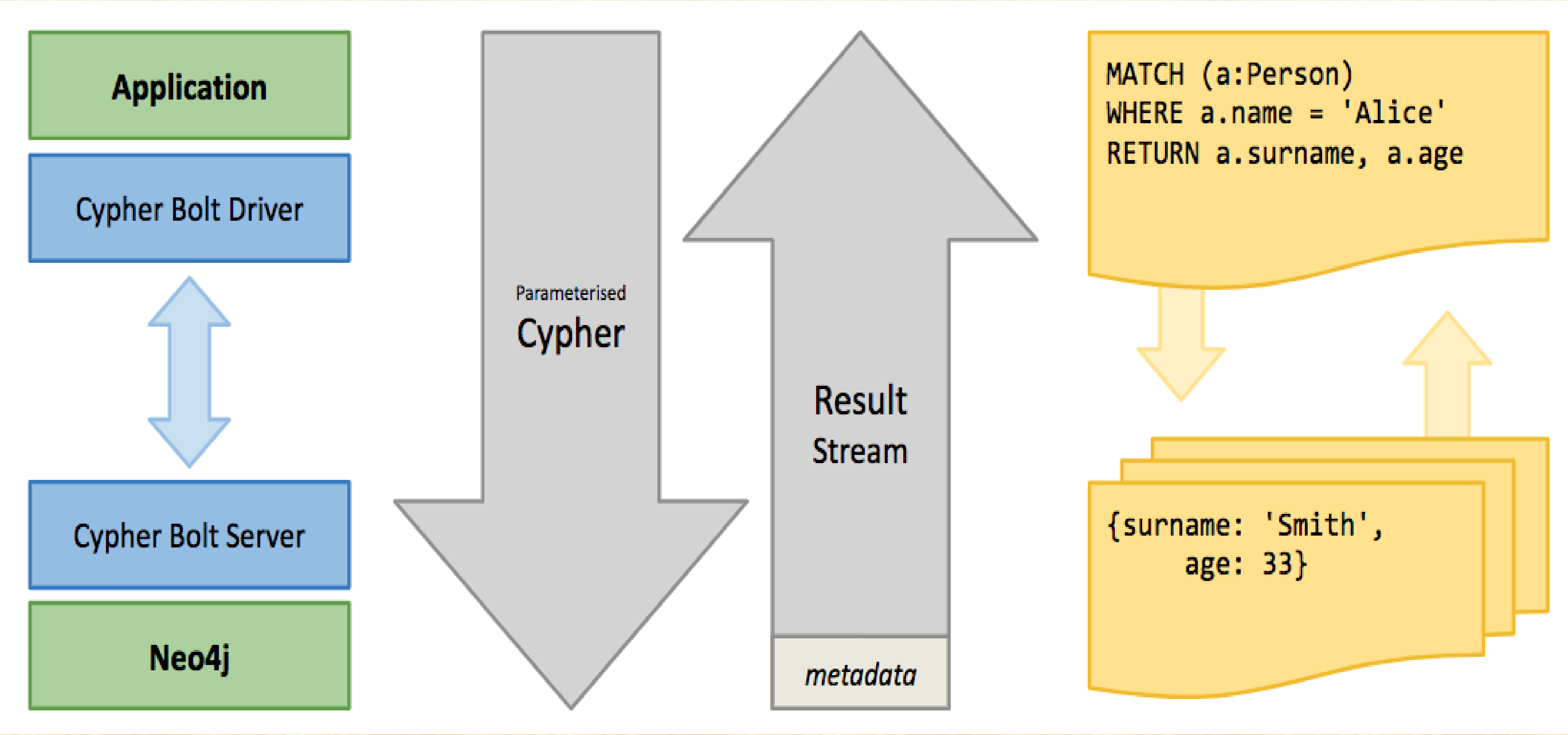
Stream of records plus metadata

Routing Drivers

- In one of the supported Neo4j drivers (Java, Javascript, Python, .Net and Go)
- there is an option to use the **bolt+routing** protocol
- E.g. URI of connection string

  bolt+routing://neo4j.myhost.com
- Routing driver decides how / where to execute the transactions/queries

How queries get run on Neo4j

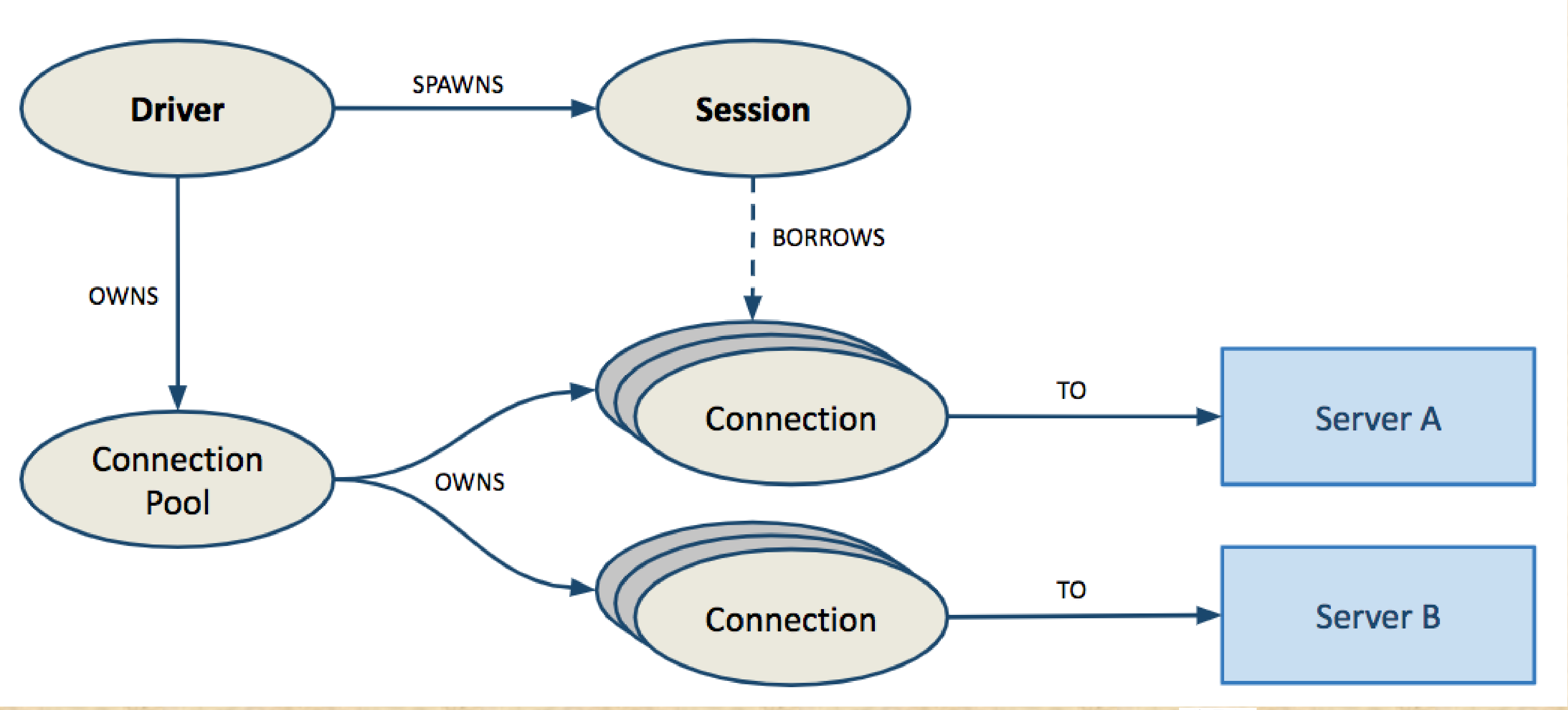


```
MATCH (a:Person)
WHERE a.name = 'Alice'
RETURN a.surname, a.age
```

```
{surname: 'Smith',
     age: 33}
```

Connection Management

- driver manages a **pool of connections** to all machines in the cluster
- user needs to just create the sessions, and run the queries from those sessions !



Installation of Neuo4j Causal Cluster on single machine

Steps

1. Download the enterprise server release from the following link

   http://neo4j.com/download/other-releases/#releases

2. Copy the downloaded compressed file in 3 separate directories that effectively creates 3 instances
3. Configure ***neo4j.conf*** file in each instance directory as follows

Modifications in *Neo4j.conf* for each instance

- dbms.backup.enabled=true
- dbms.backup.address=127.0.0.1:6362
- dbms.connector.bolt.address=127.0.0.1:7687
- dbms.connector.http.address=127.0.0.1:**7474**
- dbms.connector.https.address=127.0.0.1:7473
- dbms.mode=HA
- ha.server_id=1
- ha.initial_hosts=127.0.0.1:5001,127.0.0.1:5002,127.0.0.1:5003
- ha.host.coordination=127.0.0.1:5001
- ha.host.data=127.0.0.1:6001

Steps …

4. Start up each Neo4j instance with the following commands

   /Home directory instance1/bin/Neo4j

   /Home directory instance2/bin/Neo4j

   /Home directory instance3/bin/Neo4j

5. Startup Neo4J web based admin console in your Browser

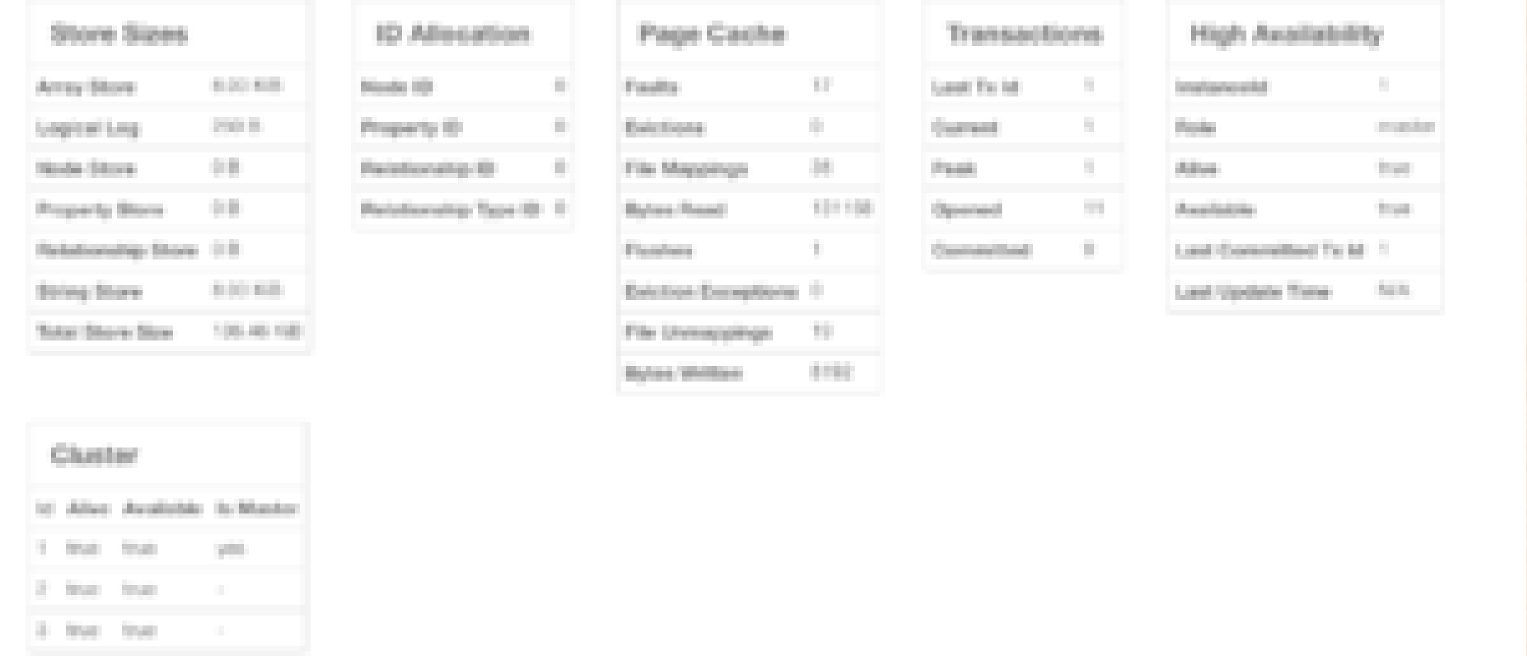
   http://127.0.0.1:7474 (instance one, HTTP)

   http://127.0.0.1:7475 (instance two, HTTP)

   http://127.0.0.1:7476 (instance three, HTTP)

Steps …

6. View the status of your cluster on the monitoring and metrics page of the Neo4J administration console



Store Sizes	
Array Store	8.00 KiB
Logical Log	[illegible]
Node Store	0 B
Property Store	0 B
Relationship Store	0 B
String Store	8.00 KiB
Total Store Size	[illegible]

ID Allocation	
Node ID	0
Property ID	0
Relationship ID	0
Relationship Type ID	0

Page Cache	
Faults	17
Evictions	0
File Mappings	[illegible]
Bytes Read	[illegible]
Flushes	1
Eviction Exceptions	0
File Unmappings	[illegible]
Bytes Written	[illegible]

Transactions	
Last Tx Id	1
Current	1
Peak	1
Opened	[illegible]
Committed	0

High Availability	
InstanceId	1
Role	master
Alive	true
Available	true
Last Committed Tx Id	1
Last Update Time	N/A

Cluster

Id	Alive	Available	Is Master
1	true	true	yes
2	true	true	-
3	true	true	-