

# Neo4j in Production

Neo Technology

[info@neotechnology.com](mailto:info@neotechnology.com) | @neo4j

derivative of "Node.Garden, Gallery of Computation" by Jared Tarbell / CC BY

# Welcome

- Find a seat. Say hi!
- If you haven't brought a computer, pair with your neighbour.
- Grab a USB key, and plug it in.

<b>Windows</b>	<b>Mac/Linux</b>
Run the "Install-Neo4j.exe"	Start a terminal
When you see Neo4j desktop	When you see the command prompt,
Click the start button	Type "./bin/neo4j start"

- Grab a coffee and get ready to begin!



# Answer five questions

- How do I deploy a Neo4j server?
- How do I performance tune it?
- How do I cluster and load balance it?
- How do I back-it up and restore it?
- How do I secure it?

# Deploy Neo4j in a Production Environment



# Schedule

Registration and initial hello

- Session 1: Deploying Neo4j
- Session 2: Back-ups, Restores, Troubleshooting
- Session 3: Security and Monitoring
- Session 4: Tuning and Performance

There will be breaks between each session, and each session includes lab exercises

# Deploying Neo4j

## What you'll learn

- Editions to choose from
- Deployment options and configurations
- Overview of the server structure
- High availability deployment

# Which Edition?

# Neo4j Editions: Community vs Enterprise

- Community Edition (GPL)
  - High Performance, fully ACID
  - Cypher, Browser
  - Server and Embedded
- Enterprise
  - Commercial Support
  - Clustering
  - Online-Backup
  - much better scaling
  - ... and many more

# When should I use Enterprise

- You want/need high availability
- You require continuous up-time
- Performance is a significant requirement
- You need a partner for support

# Deployment Options

# Two Scenarios

# Two Scenarios

## Server

- Use when want a client/server architecture

# Two Scenarios

## Server

- Use when want a client/server architecture

## Embedded

- Binding graph data management directly in your solution
- Packaged deployments

# Server Deployment

- Client/Server architecture
- Support single and HA Deployment
- REST APIs
- Client Drivers available
- Web console for admin and query

# Embedded Deployment

- Import Java library
- Supports single and HA
- Runs in your application

# Client Drivers

- Ruby
- Node
- php
- Perl
- R
- Scala

see: <http://neo4j.com/developer/language-guides>

# System Requirements

# System Requirements: Guidelines

- I/O is still a significant penalty, faster disks are better
- The more that fits into memory, the better
- JVM-based database, balance memory usage between the OS and JVM

# System Requirements: Bigger is Better

- CPU
  - Minimum: Intel Core i3
  - Recommended: Intel Core i7
- Memory
  - Minimum: 2GB
  - Recommended: 16-32GB (or more)
- Disk
  - Minimum 10GB SATA
  - Recommended: SSD w/SATA

# Bigger is Better

- File System
  - Minimum: ext4 (or similar)
  - Recommended: ext4, ZFS
- Java
  - Neo4j 2.2.x or 2.3.x: JDK 7 or 8, both Oracle or OpenJDK
- Operating System
  - Linux, HP UX, Windows 2012

# Embedded Installation

```
<project>
...
<dependencies>
  <dependency>
    <groupId>org.neo4j</groupId>
    <artifactId>neo4j-enterprise</artifactId>
    <version>2.3.0</version>
  </dependency>
  ...
</dependencies>
...
</project>
```

# Which Version?

- The Latest Version!
- Take care: milestones are not for production!

# Let's do it

## Mac OS X

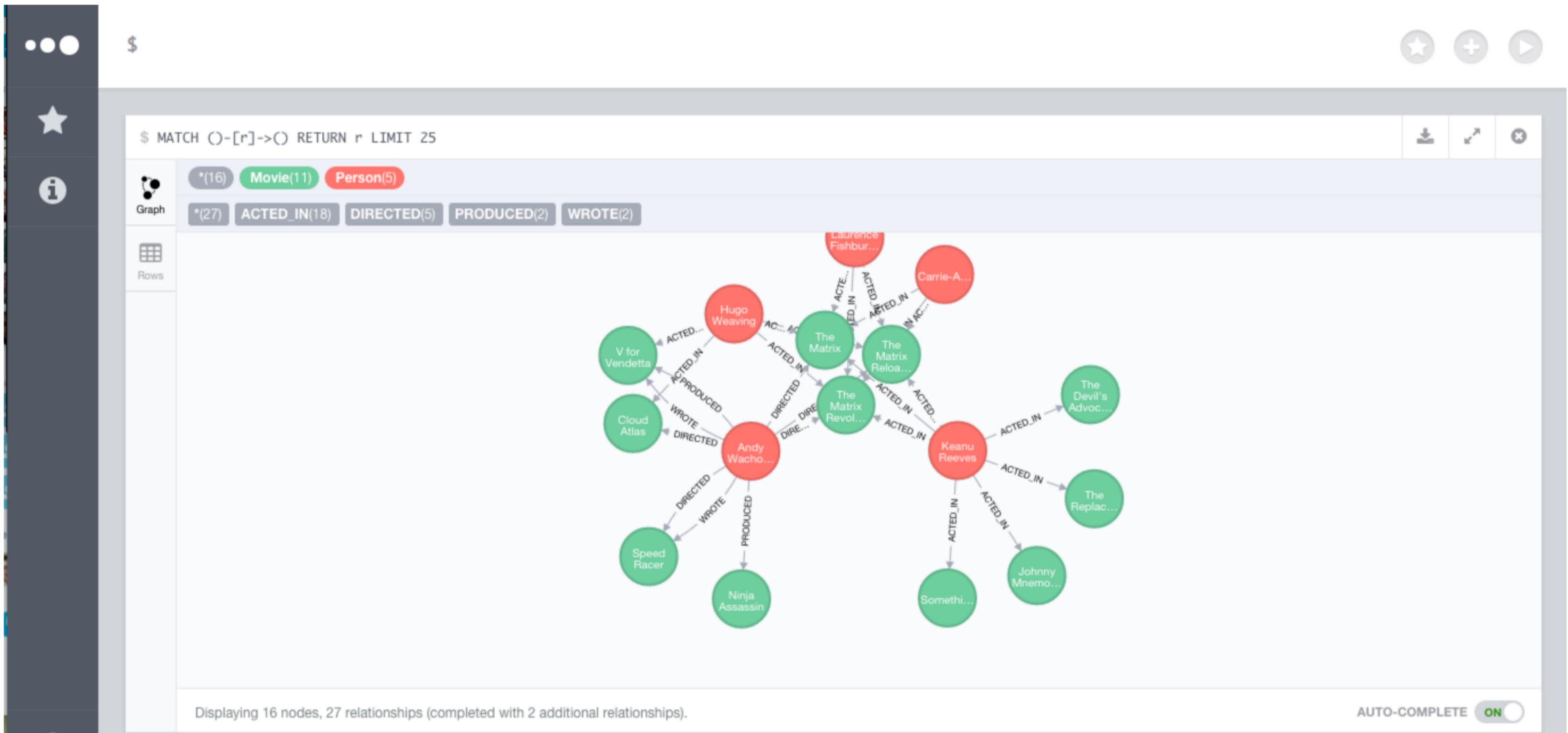
```
tar -xzf neo4j-enterprise-<VERSION NUMBER HERE>-unix.tar.gz  
neo4j-enterprise-<VERSION NUMBER HERE>/bin/neo4j start  
open http://localhost:7474/
```

## Windows

```
unzip neo4j-enterprise-<VERSION NUMBER HERE>-windows.zip  
neo4j-enterprise-<VERSION NUMBER HERE>\bin\neo4j.bat start  
start http://localhost:7474/
```

# Neo4j Basics: Get Familiar with Neo4j

# The Neo4j Data Browser: Inspect and Visualize



23 / 119



# Neo4j Browser: :play sysinfo

:play sysinfo

Store Sizes	
Array Store	24.00 KiB
Logical Log	73.76 KiB
Node Store	8.00 KiB
Property Store	15.94 KiB
Relationship Store	15.94 KiB
String Store Size	8.00 KiB
Total Store Size	2.23 MiB

ID Allocation	
Node ID	171
Property ID	383
Relationship ID	253
Relationship Type ID	6

Page Cache	
Faults	62
Flushes	19
Evictions	0
Eviction Exceptions	0
File Mappings	73
File Unmappings	57
Bytes Read	459089
Bytes Written	155436

Transactions	
Last Tx Id	18
Current	0
Peak	4
Opened	78
Committed	63

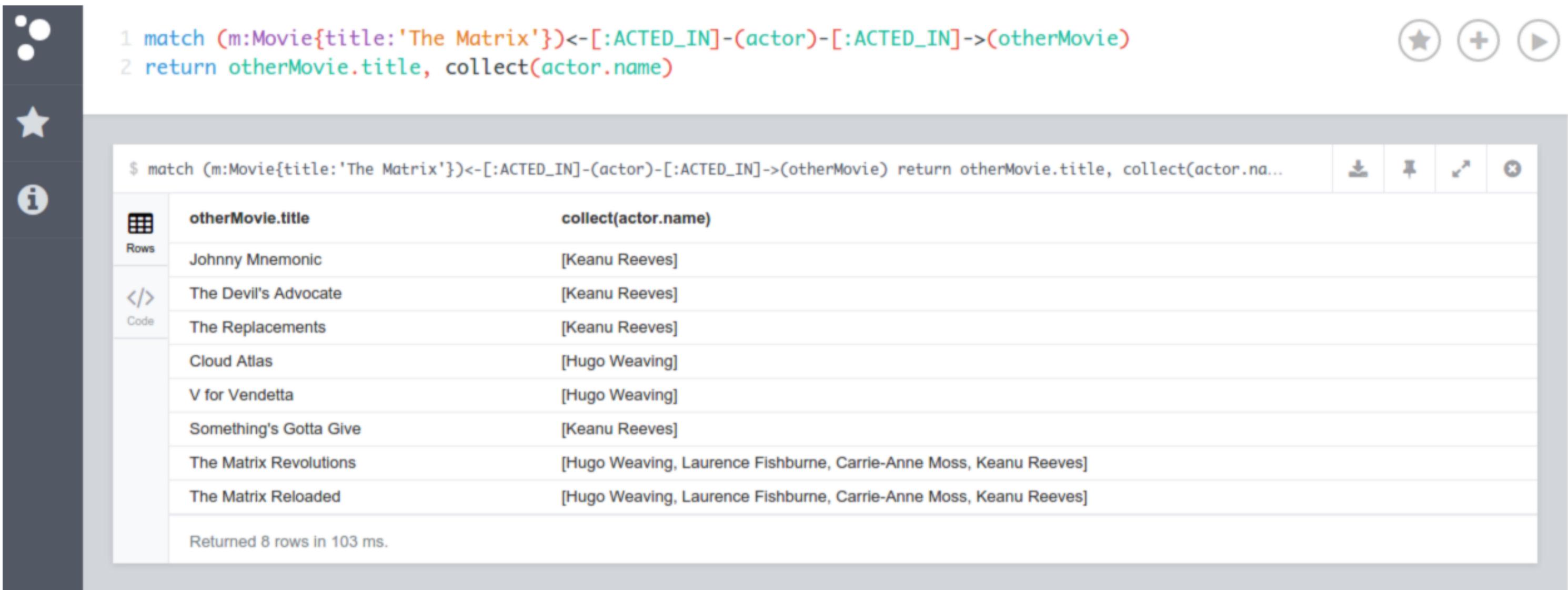
High Availability	
Not enabled.	

Cluster	
No cluster.	

Note: some metrics only available in Neo4j Enterprise.

AUTO-REFRESH  OFF

# Neo4j Browser: Running Commands and Queries



The screenshot shows the Neo4j Browser interface. On the left is a sidebar with three icons: a cluster (top), a star (middle), and an info (bottom). The main area has a toolbar with a star, plus, and play/pause buttons at the top right. Below the toolbar is a code input field containing a Cypher query:

```
1 match (m:Movie{title:'The Matrix'})->[:ACTED_IN]-(actor)-[:ACTED_IN]->(otherMovie)
2 return otherMovie.title, collect(actor.name)
```

Below the code input is a results table. The table has two columns: "otherMovie.title" and "collect(actor.name)". The table shows the following data:

	otherMovie.title	collect(actor.name)
Rows	Johnny Mnemonic	[Keanu Reeves]
</>	The Devil's Advocate	[Keanu Reeves]
Code	The Replacements	[Keanu Reeves]
	Cloud Atlas	[Hugo Weaving]
	V for Vendetta	[Hugo Weaving]
	Something's Gotta Give	[Keanu Reeves]
	The Matrix Revolutions	[Hugo Weaving, Laurence Fishburne, Carrie-Anne Moss, Keanu Reeves]
	The Matrix Reloaded	[Hugo Weaving, Laurence Fishburne, Carrie-Anne Moss, Keanu Reeves]
	Returned 8 rows in 103 ms.	

# Basic File Structure

bin/neo4j	neo4j itself
bin/neo4j-shell	a command-line shell
conf	configuration files
data/	graph data, logs
plugins/	your custom extensions (jar files)

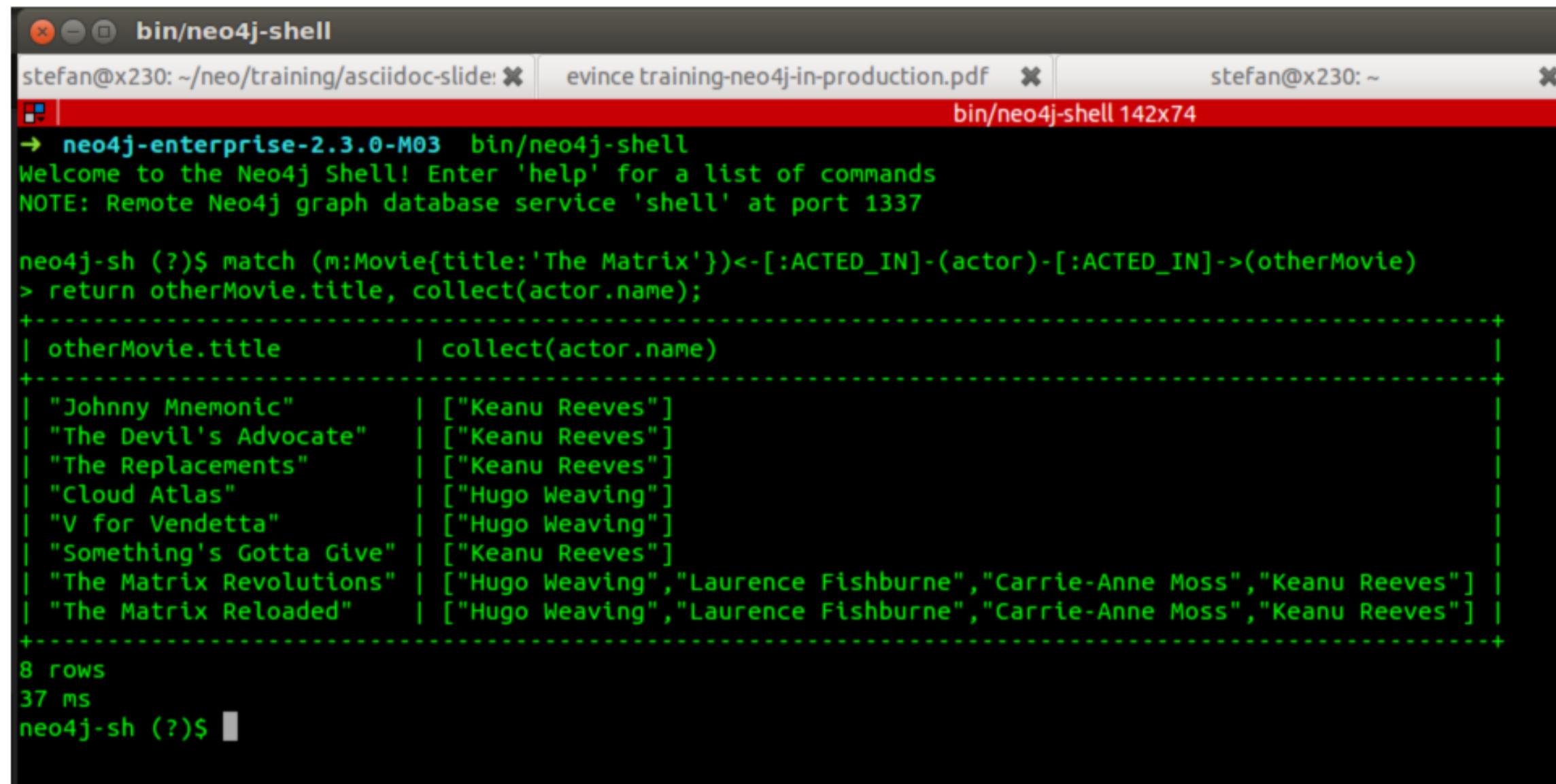
# Server Commands

bin/neo4j start	start neo4j
bin/neo4j stop	stop neo4j
bin/neo4j restart	restart neo4j
bin/neo4j status	display status (pid)
bin/neo4j info	display info (tons)

# Configuration Files

neo4j.properties	kernel configuration
neo4j-wrapper.conf	java configuration
neo4j-server.properties	server configuration
logging.properties	logging configuration

# Neo4j Shell



The screenshot shows a terminal window titled "bin/neo4j-shell" running on a Linux system. The window has three tabs at the top: "stefan@x230: ~/neo/training/asciidoc-slide", "evince training-neo4j-in-production.pdf", and "bin/neo4j-shell 142x74". The main content of the window is a Neo4j shell command and its results:

```
→ neo4j-enterprise-2.3.0-M03 bin/neo4j-shell
Welcome to the Neo4j Shell! Enter 'help' for a list of commands
NOTE: Remote Neo4j graph database service 'shell' at port 1337

neo4j-sh (?)$ match (m:Movie{title:'The Matrix'})<-[ACTED_IN]-(actor)-[:ACTED_IN]->(otherMovie)
> return otherMovie.title, collect(actor.name);
+-----+
| otherMovie.title      | collect(actor.name) |
+-----+
| "Johnny Mnemonic"   | ["Keanu Reeves"] |
| "The Devil's Advocate" | ["Keanu Reeves"] |
| "The Replacements"   | ["Keanu Reeves"] |
| "Cloud Atlas"        | ["Hugo Weaving"] |
| "V for Vendetta"     | ["Hugo Weaving"] |
| "Something's Gotta Give" | ["Keanu Reeves"] |
| "The Matrix Revolutions" | ["Hugo Weaving", "Laurence Fishburne", "Carrie-Anne Moss", "Keanu Reeves"] |
| "The Matrix Reloaded"    | ["Hugo Weaving", "Laurence Fishburne", "Carrie-Anne Moss", "Keanu Reeves"] |
+-----+
8 rows
37 ms
neo4j-sh (?)$
```

# Take a few minutes to Browse

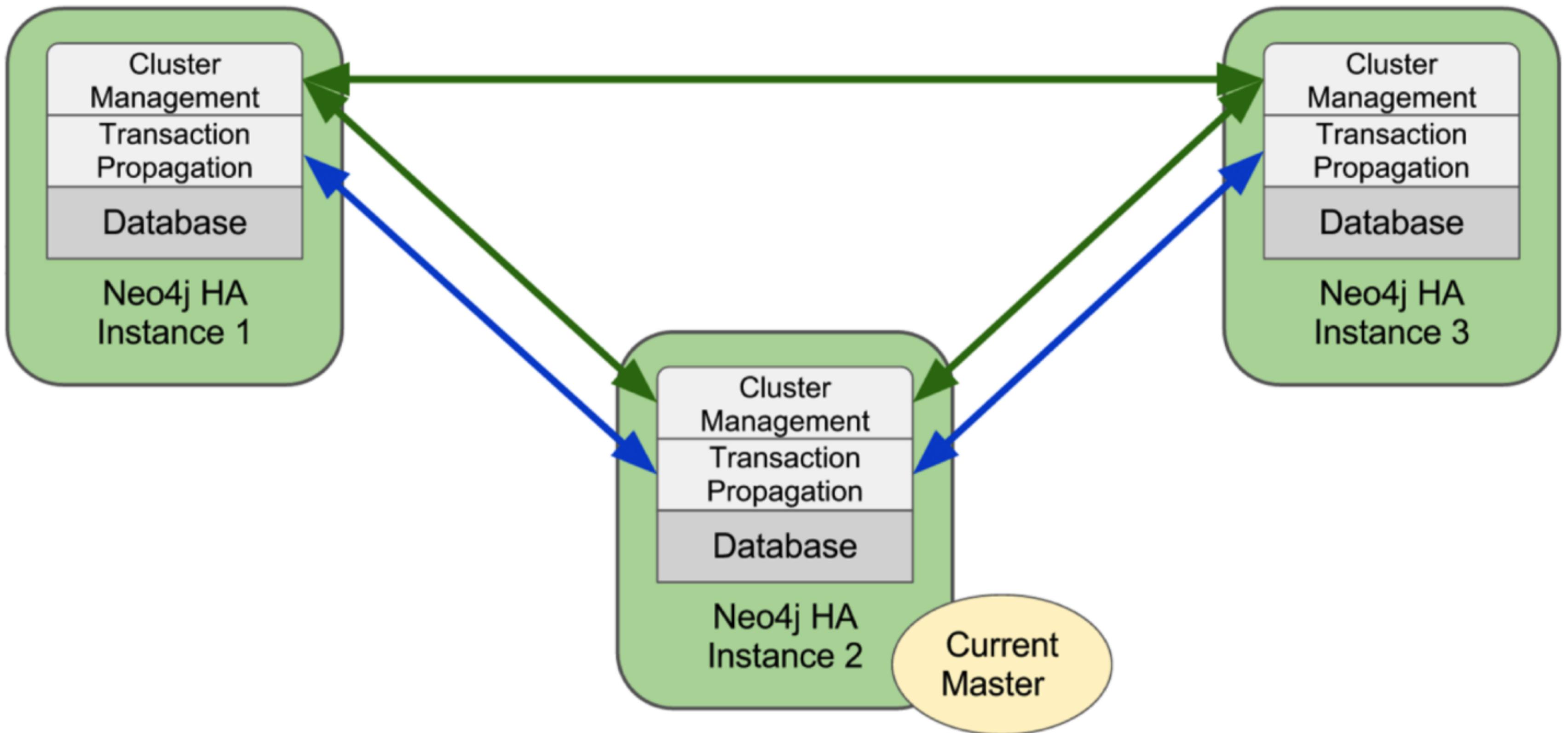
- Start and stop the server
- Check out the movie graph
- Use the browser
- Browse the directory
- Get some coffee

# Cluster Neo4j

# Benefits

- Fault Tolerance in the event of a failure
- Horizontally Scalable Reads
  - Load Balanced Requests
  - Cache Sharding

# Architecture



# Neo4j Cluster

- Single Master
- Slaves synchronize Writes with Master
- Master can push writes to Slaves
- Slaves can pull writes from Master

# Writes

On Slaves:

- will communicate with Master and wait until Master commits

On Master:

- will optimistically push to slaves on commit

# Enable HA

In neo4j-server.properties:

org.neo4j.server.database=HA

# Cluster Membership

- Uses Paxos implementation
- Must have  $\geq 3$  Members for Failover ( $2n+1$ )
- an odd number of members is recommended
  - Need a quorum ( $> 50\%$  of cluster to do master election)
- Allows for Arbiter Instances: **neo4j/bin/neo4j-arbiter start**

# Joining the Cluster

Knowing cluster members:

- Set `ha.server_id` to a cluster-wide unique value
- Set

`ha.initial_hosts=server1:5001,server2:5001,server3:5001`

# Cluster Guidelines

- Proxy to load balance
- Decide write policy (master/slaves)
- Elections
- Push vs. Pull replication
  - ha.tx\_push\_factor=1
  - ha.pull\_interval=10
  - ha.tx\_push\_strategy=fixed
- Slave\_only configuration

# Cluster Deployment Scenarios

- LAN deployment
  - straight forward → what we do right now.
- WAN deployment
  - primary DC: normal setup
  - secondary DCs: slave only
  - if primary DC burns down: reconfigure instance in secodary to not slave only

# Cluster Upgrades

- shut down slaves
- shut down master
- upgrade master in single mode (**allow\_store\_upgrade=true**)
- stop master
- start master in HA mode
- upgrade slaves, remove old datastore (**data/graph.db**)
- start slaves

# Loadbalancers

## Challenges:

- send writes to master
  - how to distinguish r/w ?
- distribute reads over slaves (or all)
- stickyness for tx cypher endpoint
- auth for keepalive checks

<http://blog.armbruster-it.de/2015/08/neo4j-and-haproxy-some-best-practices-and-tricks/>

# Haproxy 1.6 sample config snippet

```
defaults
    mode http
    ...

frontend http-in
    bind *:8090
    acl write_method method POST DELETE PUT
    acl write_hdr hdr_val(X-Write) eq 1
    acl write_payload payload(0,0) -m reg -i CREATE|MERGE|SET|DELETE|REMOVE
    acl tx_cypher_endpoint path_beg /db/data/transaction
    http-request set-var(txn.tx_cypher_endpoint) bool(true) if tx_cypher_endpoint
    use_backend neo4j-master if write_hdr
    use_backend neo4j-master if tx_cypher_endpoint write_payload
    use_backend neo4j-all if tx_cypher_endpoint
    use_backend neo4j-master if write_method
    default_backend neo4j-all

backend neo4j-all
    option httpchk GET /db/manage/server/ha/available HTTP/1.0\r\nAuthorization:\ Basic\ bmVvNGo6MTIz
    acl tx_cypher_endpoint var(txn.tx_cypher_endpoint),bool
    stick-table type integer size 1k expire 70s # slightly higher with org.neo4j.server.transaction.timeout
    stick match path,word(4,/) if tx_cypher_endpoint
    stick store-response hdr(Location),word(6,/) if tx_cypher_endpoint
    server s1 127.0.0.1:7474 maxconn 32 check
    server s2 127.0.0.1:7475 maxconn 32 check
```



# Setup our Cluster

- Stop Running Instance
- Make two copies of Neo4j Directory
- Follow Instructions on:

<http://neo4j.com/docs/stable/ha-setup-tutorial.html#ha-local-cluster>

# Break

# Backups, Restores, Troubleshooting

## What you'll learn

- Backup process and application
- Restoring Neo4j from backup
- Troubleshooting issues

# Backup Basics

`/bin/neo4j-backup` utility

One utility for

- Full Backup
- Incremental

Use scripts/jobs to automate

# Full Backup

- Copies Files
- Does not take locks
- Replays transactions ran after back-up started until end of copy
- invokes consistency check

# Full Backup (try it)

Create a blank directory **/mybak**

```
./neo4j-backup -from 127.0.0.1 -to ./mybak
```

What about backup from HA?

- choose one slave and use it

# Incremental Backup

- Replays logs of Transactions since last backup
- Requires full backup be done first
- Backup tool auto detects full or incremental
- invokes consistency check

# Incremental Backup (try it)

Create a new node first (so you have a new transaction)

```
./neo4j-backup -from 127.0.0.1 -to ./mybak
```

# Backup Guidelines

- Frequent incremental backups for high transaction applications
- Backup from same instance
- Consider designating a slave for backups in clustered deployment

# Restores

# Really Simple (3 steps)

- Make sure Neo4j is not running first
- Replace `graph.db` with the backup
- Start Neo4j

# Backup Guidelines

- Daily/Weekly Full Backups
- Incremental every X hours
- Keep a copy of Snapshot around
- Conform to your IT policies
- point in time recovery: use your existing enterprise backup system

# TroubleShooting

# Logical Log Files: Limit em!

```
keep_logical_logs=true          // NO! You'll run out of space!  
keep_logical_logs=10 files     // Yes  
keep_logical_logs=300M size    // Yes  
keep_logical_logs=250k txs     // Maybe  
keep_logical_logs=24 hours     // Maybe (depends on Incr. Backup)  
keep_logical_logs=30 days      // Maybe (depends on Incr. Backup)
```

# Logical Log Files

Logical logs are needed for

- Incremental backups
  - Cluster transaction propagation
- align time period with backup frequency and hardware SLAs

# Garbage Collection Logging

Turn it on right now!

```
neo4j-wrapper.conf  
# Uncomment the following lines to enable garbage collection logging  
wrapper.java.additional=-Xloggc:data/log/neo4j-gc.log  
wrapper.java.additional=-XX:+PrintGCDetails  
wrapper.java.additional=-XX:+PrintGCDateStamps  
wrapper.java.additional=-XX:+PrintGCAfterApplicationStoppedTime
```

# Garbage Collection

```
tail -f neo4j-gc.log | grep Full
```

```
2013-02-08T11:44:08.493-0600: 1518.855: [Full GC 2628247K->2598251K(3728320K), 6.3689364 secs]
2013-02-08T11:44:15.593-0600: 1525.956: [Full GC 2997119K->2750610K(3728320K), 6.8108610 secs]
2013-02-08T11:44:23.248-0600: 1533.613: [Full GC 3216722K->2771560K(3728320K), 7.8142391 secs]
2013-02-08T11:44:31.902-0600: 1542.269: [Full GC 3237672K->2965288K(3728320K), 6.7392397 secs]
```

Oh no!

# What if GC shows long pauses?

- You'll need to do some configuration tuning (coming later)
- Full GC can cause application pauses, master failover

# Index Sizes: Check them

```
ls -alh neostore*
382M Jul 15 20:36 neostore.propertystore.db
```

```
graph.db/schema# du -sh .
138 G
```

# Consistency Check

On offline copy of store (like say backup)

```
java -cp 'lib/*:system/lib/*' \
org.neo4j.consistency.ConsistencyCheckTool <your datastore dir>
```

# Cluster Inconsistency: Branching

- Failed master returns to the cluster
- Its database is inconsistent from cluster at the time of the failure
- Neo4j “branches” that database
- Returning instance uses cluster DB

# Branched Data

- `ha.branched_data_policy=keep_all`
- `ha.branched_data_policy=keep_last`
- Never ever ever do this: `ha.branched_data_policy=keep_none`

# Branched Data

- **data/graph.db/branched** directory
- It is a copy of the database, can be opened, queried, etc.
- No automatic reconciliation will happen

# Break

# Security and Monitoring

## What you'll learn

- Current capabilities
- Securing server access
- Monitoring techniques and tools

# Security

# Data Collector



# Data Collector

## Neo4j Usage Data Collector

- pings udc.neo4j.org

No Personal information

- Edition, OS, Java, REST client

To disable

- `neo4j.ext.udc.disabled=true`
- Remove `neo4j-udc-* .jar`
- Put in Exclusion in pom.xml

# Web Server

# Localhost by default

conf/neo4j-server.properties

- org.neo4j.server.webserver.port=7474

Uncomment for all addresses:

- #org.neo4j.server.webserver.address=0.0.0.0

# HTTPS Support

## conf/neo4j-server.properties

```
org.neo4j.server.webserver.https.cert.location=(your cert file)  
org.neo4j.server.webserver.https.key.location=(your key file)
```

Make sure Neo4j can read/write private key.

# Arbitrary Code

- Traversal Endpoint
  - Uses Javascript under the hood
- NeoShell
  - jsh command

# Arbitrary Code (mitigate)

- Secure Traversal Endpoint with Server Authorization Rules
  - See [https://github.com/maxdemarzi/neo\\_lock\\_down](https://github.com/maxdemarzi/neo_lock_down)
- In `neo4j.properties`:
  - `remote_shell_enabled=false`
- In `neo4j-server.properties`:
  - `org.neo4j.server.manage.console_engines=none`

# Server authentication

- enabled by default
- no authorization (yet)

# Server Authorization Rules

- You'll have to write some Java
  - Block any URL
  - Connect to existing Authorization systems (ex. LDAP)
  - Works on Unmanaged Extensions/Plugins too

# Security Limitations

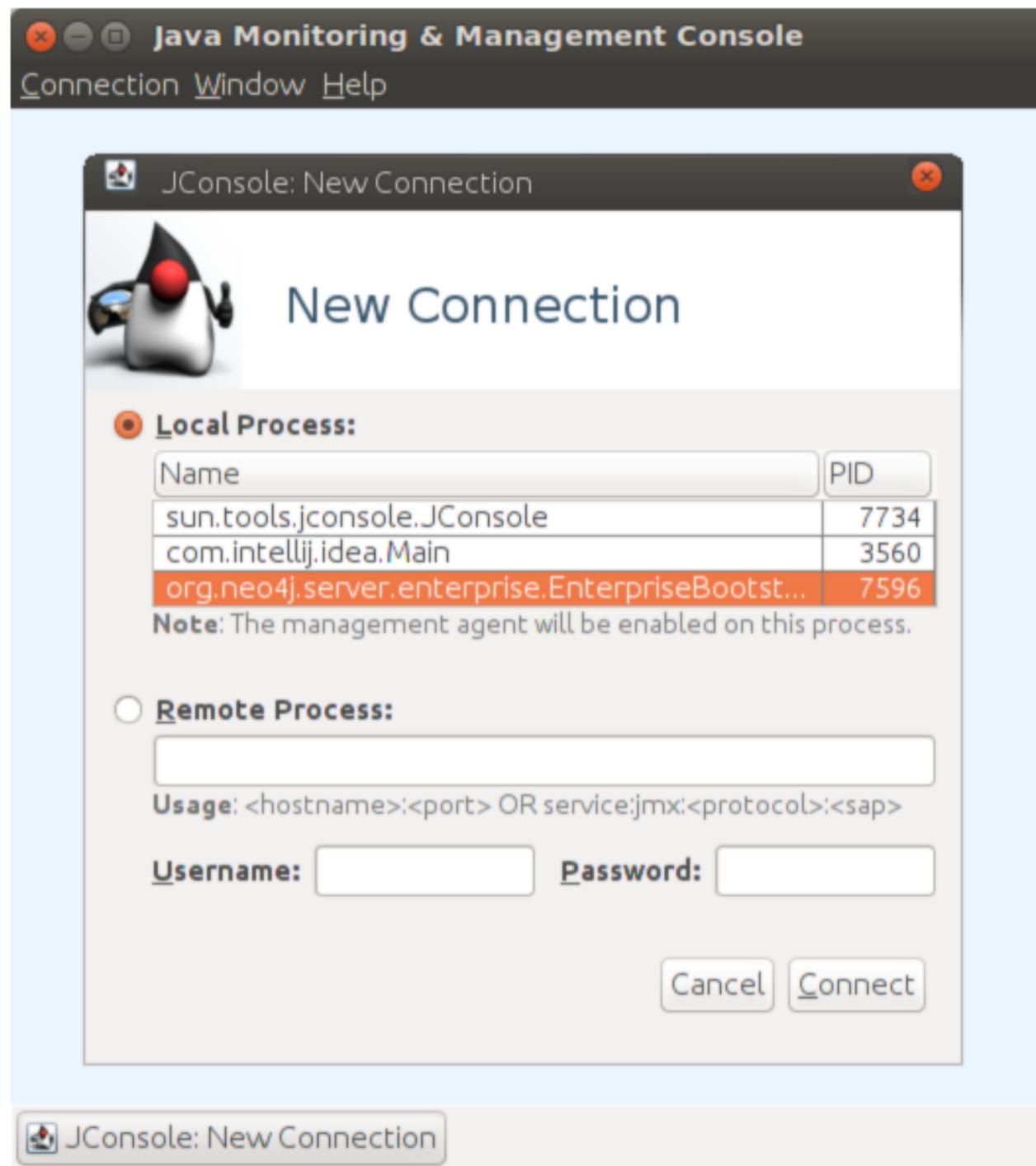
- Access Control
- Node/relationship level security is coming
- Control how Neo4j is exposed to outside world - use a firewall or proxy

# Monitoring

# Java Management Extensions(JMX)

- Use JConsole or your favorite JMX capable monitoring tools
- Try it: type “jconsole”
  - Connect to `org.neo4j.server.[Enterprise]Bootstrapper`

# JConsole



You can use Insecure Connection when on localhost

# JMX MBeans

- Many Mbeans available for monitoring
- See documentation: “Reference of supported JMX Mbeans”

Configuration	Diagnostics
High Availability	Kernel
Locking	PageCache
Store Files Sizes	Transactions

# Remote Monitoring

```
neo4j-wrapper.conf:  
# Remote JMX monitoring, uncomment and adjust the following lines as needed.  
#wrapper.java.additional=-Dcom.sun.management.jmxremote.port=3637  
#wrapper.java.additional=-Dcom.sun.management.jmxremote.authenticate=true  
#wrapper.java.additional=-Dcom.sun.management.jmxremote.ssl=false  
#wrapper.java.additional=-Dcom.sun.management.jmxremote.password.file=conf/jmx.password  
#wrapper.java.additional=-Dcom.sun.management.jmxremote.access.file=conf/jmx.access  
# Some systems cannot discover host name automatically, and need this line configured:  
#wrapper.java.additional=-Djava.rmi.server.hostname=$THE_NEO4J_SERVER_HOSTNAME
```

# Monitoring by neo4j-shell

```
echo 'dbinfo -g "High Availability" InstancesInCluster' | bin/neo4j-shell
```

# metrics plugin

- send continuously metrics info to
  - csv file
  - graphite
  - ganglia

# Coffee

# Tuning and Performance

## What you'll learn

- Internals and caches
- Configuration tuning guidance
- Performance guidance

# Tuning

# Neo4j Caches

## PageCache (off heap)

- Same format as storage

## Object Cache ( gone away since 2.3 )

- Formatted for high speed traversals

# PageCache settings

- Try to map as much of your store into page cache as possible
  - By default, Neo4j assigns 50% of (RAM – heap) to page cache
  - Calculate size: sum all **store.db** files + anticipated growth

```
dbms.pagecache.memory=10g  
dump_configuration=true
```

# Calculating File Sizes

number\_of\_nodes \* 15 bytes

- 10M nodes = 90M

number\_of\_relationships \* 34 bytes

- 100M relationships = 3G

# Disable Access Time Updates (Linux only)

- Optimize for small transaction High IO load:
- Add to options on **/etc/fstab**:
  - noatime,nodiratime

# Number of Open Files

- Change default from 1024 to 40000
- Edit `/etc/security/limits.conf`  
`(/etc/security/limits.d/neo4j on Ubuntu)`
  - neo4j soft nofile 40000
  - neo4j hard nofile 40000
- Edit `/etc/pam.d/su` and add:
  - session required pam\_limits.so
- Restart System

# Heap Size

## *Guidelines for heap size*

<b>Number of primitives</b>	<b>RAM size</b>	<b>Heap configuration</b>	<b>Reserved RAM for the OS</b>
10M	2GB	512MB	the rest
100M	8GB+	1-4GB	1-2GB
1B+	16GB-32GB+	4GB+	1-2GB

In neo4j-wrapper.conf:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=4096
# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=4096
```

# Linux I/O Scheduler

- Use NOOP or Deadline
- Avoid CFQ

```
echo noop > /sys/block/sda/queue/scheduler  
cat /sys/block/sda/queue/scheduler  
[noop] deadline cfq
```

# Storage

- use SSD Drives
- 15k sas-2 w/ hardware RAID 10 configuration
- specialized hardware for specific need (e.g. FusionIO)

# Ondisk data layout

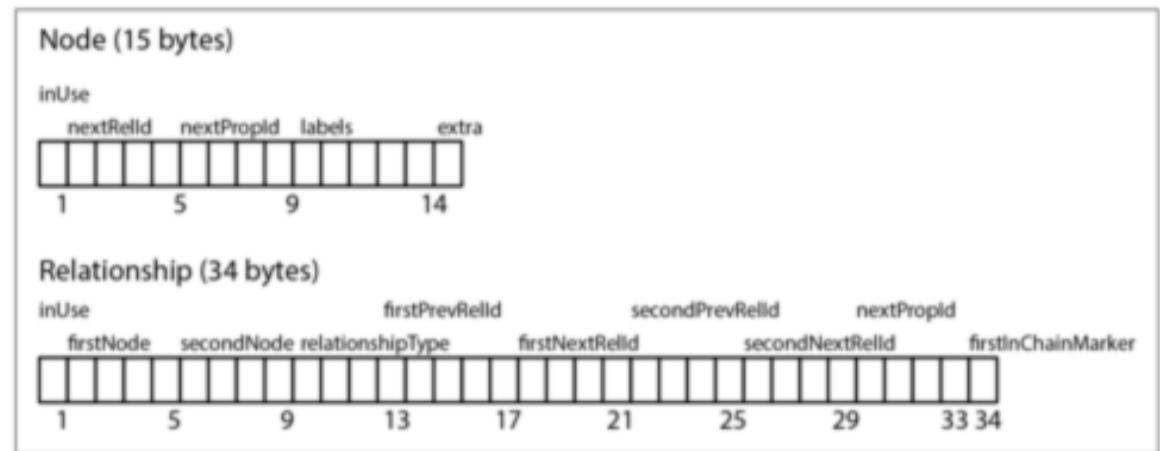


Figure 6-4. Neo4j node and relationship store file record structure

# Short Strings

## *Character count limits*

String class	Character count limit
Numerical, Date and Hex	54
Uppercase, Lowercase and E-mail	43
URI, Alphanumerical and Alphasymbolical	36
European	31
Latin1	27
UTF-8	14

# Short Numeric Arrays

- Based on highest leftmost set bit
- Less than 24 bytes after compression
- Less than 64 members
- Smaller numbers are better... until negative.
  - [0L,1L,2L,4L] = 12 bits (Yes!)
  - [-1L,1L,2L,4L] = 32 bytes (No! “-1” uses 64 bits)

# Performance

# Performance Basics

- Cypher is fast, and getting faster
- Java is really fast
  - Millions of traversals per second
  - Thousands of writes per second

# JVM Tuning

Ensure the heap size is set to an appropriate size

- Too little or too big can be a problem

Use G1 garbage collector

- `-XX:+G1GC`
- `-XX:MaxGCPauseMillis=200`

# Explain/Profile a Query

- put **EXPLAIN** before your Cypher statement:
  - query plan with estimated counts is shown
- put **PROFILE** before your Cypher statement:
  - execution statement + query plan with exact counts is shown
- Ensure indexes are being used

# Data Import

- **LOAD CSV** for online data loads
  - If write then read, ensure indexes are also created during loading
- for initial data loading: **neo4j-import**
- Java based batch inserter API

# Checkpointing

- Flush and force store files
- Ensures log rotations are short
- Long log rotations in 2.2.x could cause pauses
- By default, checkpointing occurs every 5 mins, or every 100,000 transactions, whichever comes first

```
dbms.checkpoint.interval.time=2m
```

```
dbms.checkpoint.interval.tx=10000
```

# Some Final Production Thoughts

- Simulate your production workload before production rollout
- Test HA, failover, fallback, and backup policy before Production rollout
- Production is not the place to tune performance

## Learn More about Using Neo4j

<http://neo4j.com>

# Graph Academy

<http://graphacademy.com>

- Online Courses
- Classroom Training like this one
- On Site Brown-Bag Lunches (Lunch & Learn)
- Webinars
- Graph Days (free)
- Graph Connect Conferences

# Neo4j Reference Manual

- <http://neo4j.com/docs/stable>
- <http://neo4j.com/docs/stable/cypher-refcard>

Let's have a look

# Neo4j Books

<http://neo4j.com/books>

- Graph Databases (free PDF)
  - O'Reilly: I.Robinson, J.Webber, E.Eifrem
- Learning Neo4j (free PDF) & 6 more
  - Packt: Rik van Bruggen
- Practical Neo4j
  - Apress: Greg Jordan
- many more ...

# Talk to us!

<http://neo4j.com/contact-us>

- Have an interesting project, use-case?
  - We can help you promote it with articles, talks, webinars
- Wrote a tool, library, connector?
  - We love to make it known in our community
- Have questions about licensing & pricing?
  - Reach out with your questions, we help you clarify
- Want to work on Neo4j?
  - We offer many cool jobs, please apply

# Business & Production Offers

<http://neo4j.com/professional-services/>

- Neo4j Enterprise (faster, scalable, supported)
- Service & Support (24/7 and other SLAs)
- Workshops, On-Site Training, Proof of Concept
- Dedicated Consulting
- Huge Partner Network
- We don't bite, ping your local rep for a relaxed chat

# Get Involved in the Neo4j Community

# GraphGists: Share your Graph

<http://graphgist.neo4j.org>

- Have a great idea / use-case for Neo4j?
- Model it as a GraphGist
- Submit it to our GraphGist Portal
- Share on Twitter and elsewhere
- Participate in our regular challenges!

# Neo4j Developer Pages

<http://neo4j.com/developer>

- Intro to Graphs, RDBMS to Graphs
- Data Import & Data Modeling
- Language Guides & Drivers
- Visualization
- Neo4j Ecosystem

# Ask Questions ... Get Helpful Answers

## Stack Overflow



Find answers or reach to fellow developers with questions.

[Ask Neo4j questions »](#)

## GitHub Issues



Encountered an issue with Neo4j?  
Submit it here.

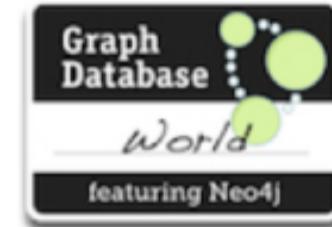
## Neo4j Google Group



Share your experiences and expertise with fellow graphistas.

[Join now »](#)

## Meetups / User Groups



Neo4j meetups are worldwide. Make a connection or start a new group.

[Join a Meetup »](#)

# Ask Questions ... Get Helpful Answers

## Stack Overflow



Find answers or reach to fellow developers with questions.

[Ask Neo4j questions »](#)

## GitHub Issues



Encountered an issue with Neo4j?  
Submit it here.

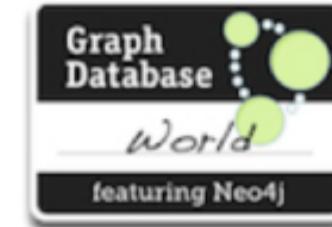
## Neo4j Google Group



Share your experiences and expertise with fellow graphistas.

[Join now »](#)

## Meetups / User Groups



Neo4j meetups are worldwide. Make a connection or start a new group.

[Join a Meetup »](#)

## Quick Links

- <http://stackoverflow.com/questions/tagged/neo4j>
- <http://github.com/neo4j/neo4j/issues>
- <http://groups.google.com/group/neo4j>
- <http://neo4j.meetup.com/>



# Thank You

Neo Technology

[info@neotechnology.com](mailto:info@neotechnology.com) | @neo4j