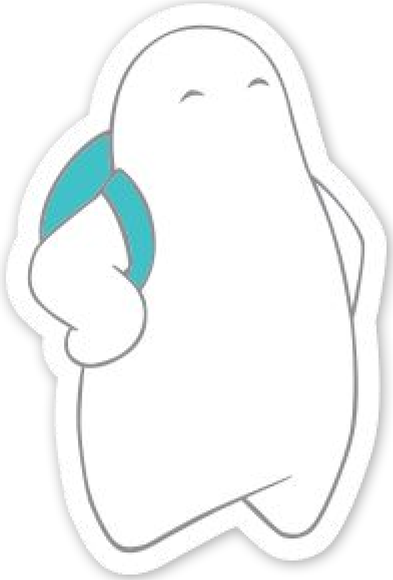
404 SQL DB not found. Meet RethinkDB



Who am I ?

My name is Nir Galon, I’m a Python developer, work at gizra, very passionate about open source and the web.

GitHub: nirgn975

Twitter: @nirgn975

Website: nirgn.com

Blog: lifelongstudent.net

What we'll talk about ?

1. What is RethinkDB ?

RethinkDB

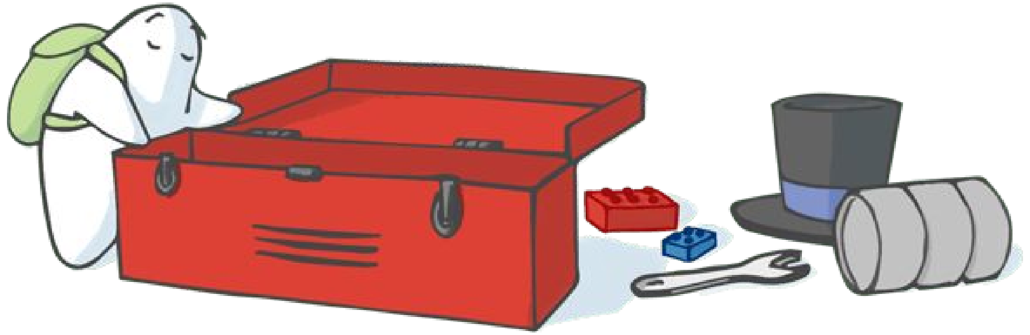
2. How it differs from MongoDB ?

mongoDB®

ReQL

We used ReQL for querying — a custom query language that provides a clear syntax that used to manipulate JSON documents.

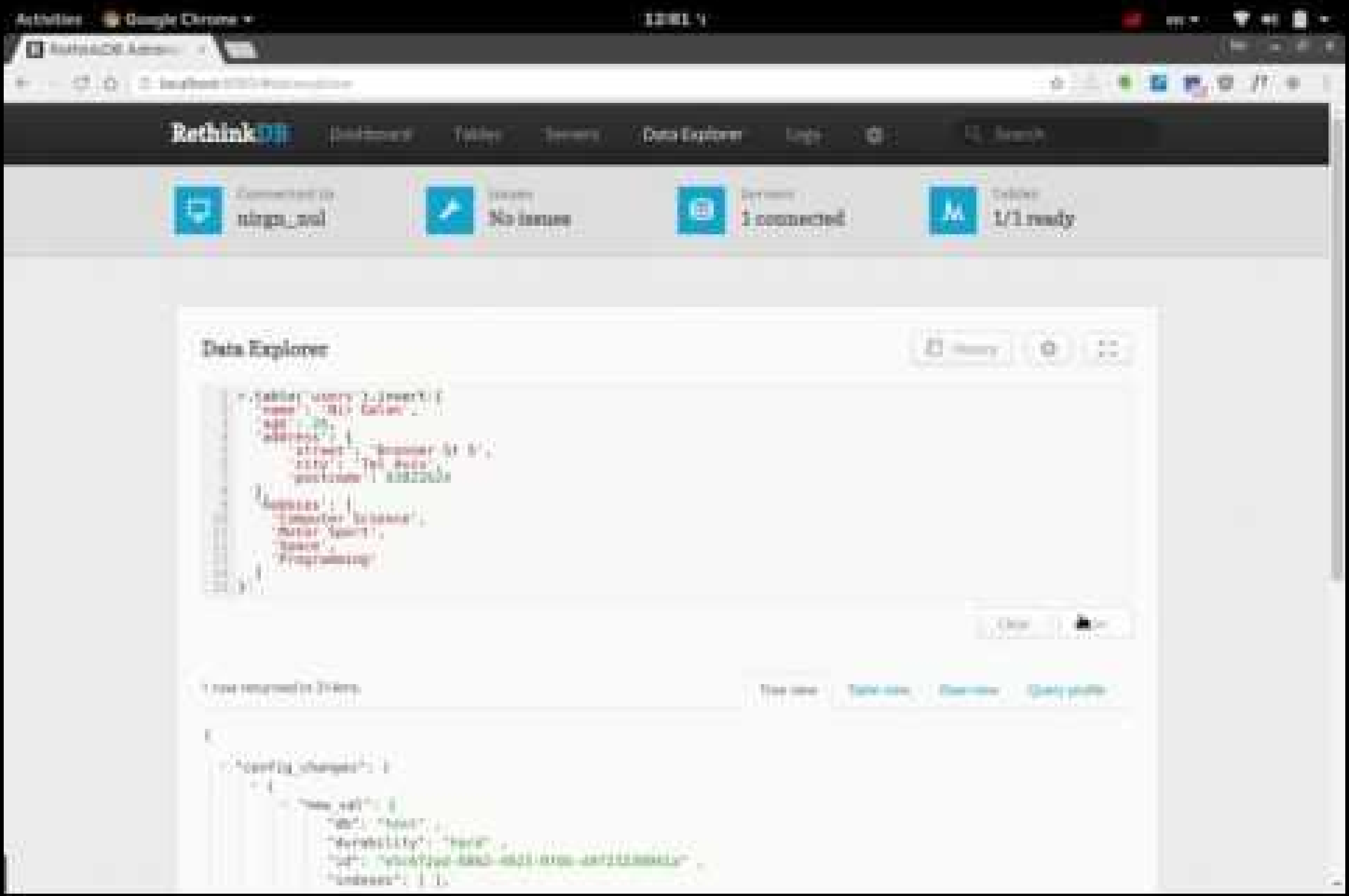
- r.table(‘users’).filter({ name: ‘Nir’ }).orderBy(r.desc(‘age’))
- r.table(‘users’).pluck(‘city’).distinct().count()
- r.table(‘users’).filter(r.row(‘name’).eq(‘nir’)).update({“name”: “nirgn”})



Let's start with the basics

What we’ll do:

- Activate RethinkDB instance.
- Check the admin panel, at localhost:8080.
- See the default `test` db that created for us.
- Create a new table named `users`.
- Create a documents (rows) with couple of fields.
- Run a simple Query to find the `postcode`.
- Show the table, and show a document.



Data Explorer

```
.table('users').insert({
 'name' : 'Ali Gulan',
 'age' : 25,
 'address' : {
      'street' : 'Brenner St 5',
      'city' : 'Tel Aviv',
      'postcode' : [illegible]
 },
 'hobbies' : [
   'Computer Science',
   'Motor Sport',
   'Space',
   'Programming'
 ]
})
```

1 row returned in [illegible] ms

```
{
  "config_changes": [
    {
      "new_val": {
          "db": "test" ,
          "durability": "hard" ,
          "id": "[illegible]" ,
          "indexes": [ ],
```

Let’s go over RethinkDB main features

1. The most ground breaking change is: **Changefeeds**

```
r.table('games').filter({‘game’: ‘chess’}).orderBy('score').limit(3).changes()
```

Real world:

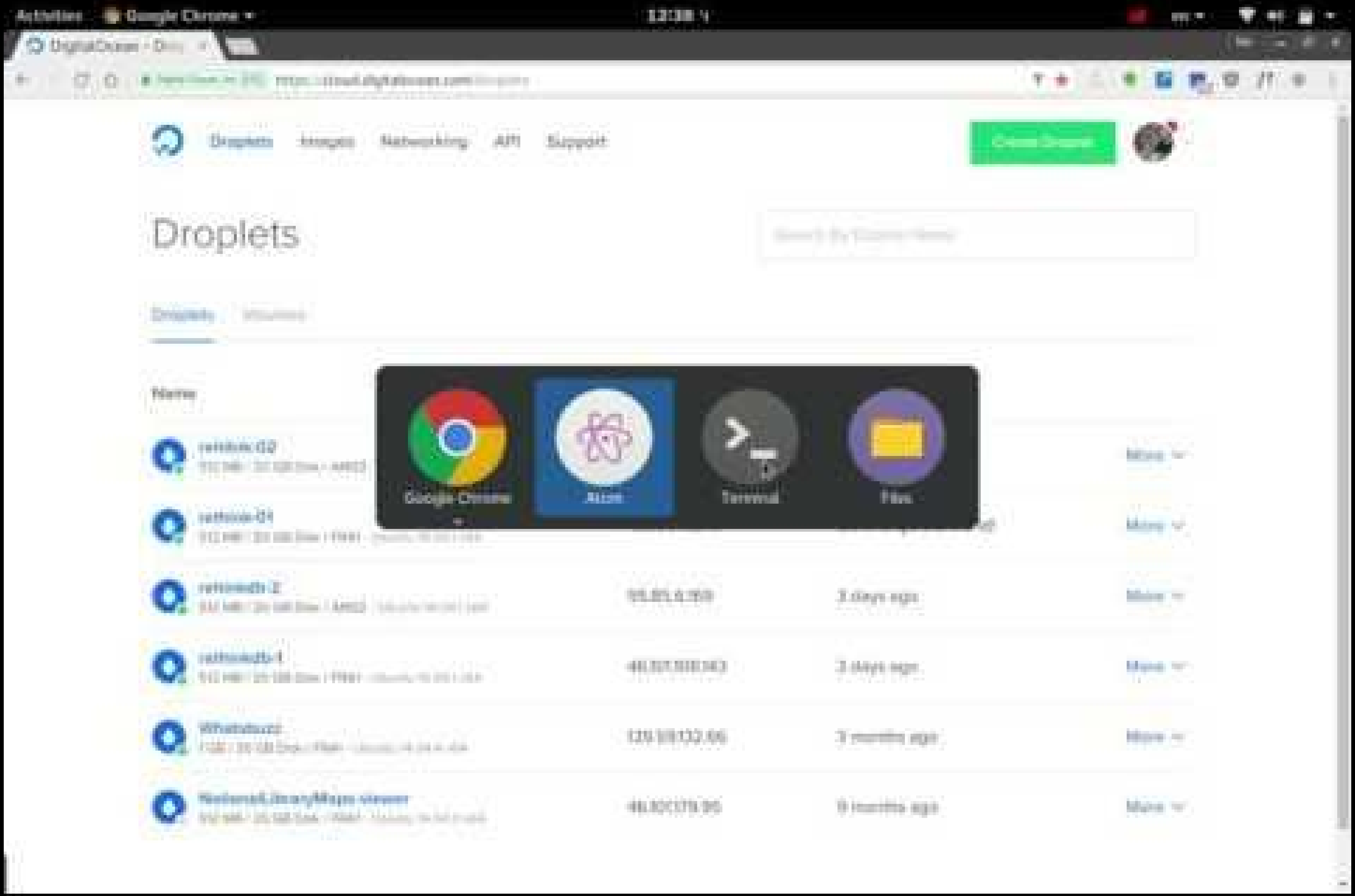
```
r.table('games').filter({‘game’: ‘chess’}).orderBy('score').limit(3)
             .changes().run(conn, callback);
```

2. Scalability - Clustering

Clustering in RethinkDB means a couple of instance operate as a single DB. It help us achieve data scalability, load balance, and increase tolerance.

What we'll do:

- Create RethinkDB cluster.
- Configure an instance to be a part of the cluster.
- Add new machines to existing cluster.
- Look how the admin interface displays the status of the cluster and signals any problems.



Droplets

Droplets Volumes

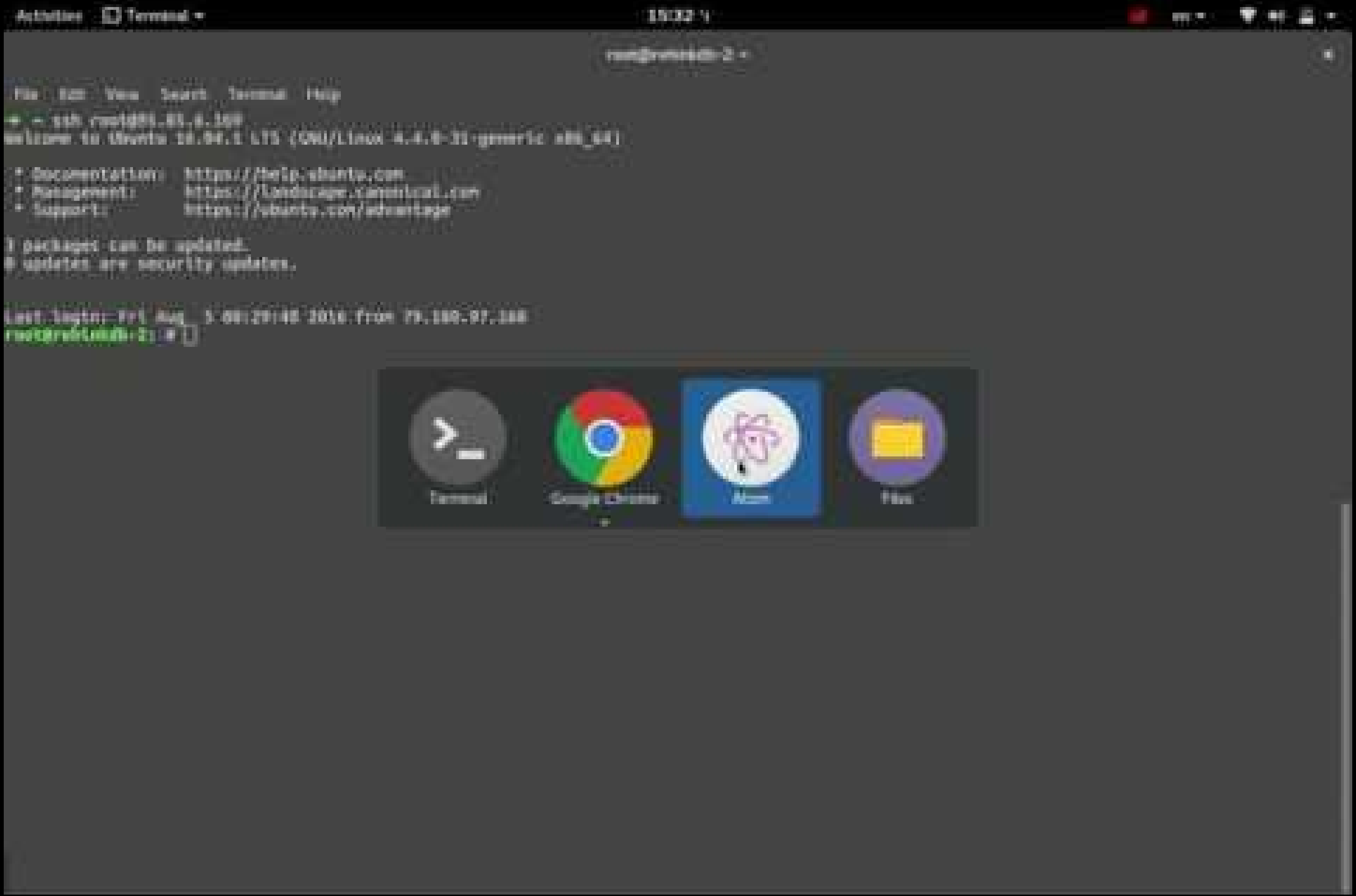
Name			
[illegible]	[illegible]	[illegible]	More
[illegible]	[illegible]	[illegible]	More
[illegible]	95.85.6.159	3 days ago	More
[illegible]	[illegible]	3 days ago	More
Whatsbuzz	128.199.132.66	3 months ago	More
NationalLibraryMaps-viewer	[illegible]	9 months ago	More

2. Scalability - Replication

Replication mean to save couple of copies of our data. It helps us increase redundancy and data availability.

What we'll do:

- We'll set up a two member replica set - because we have only 2 servers connected to our cluster.



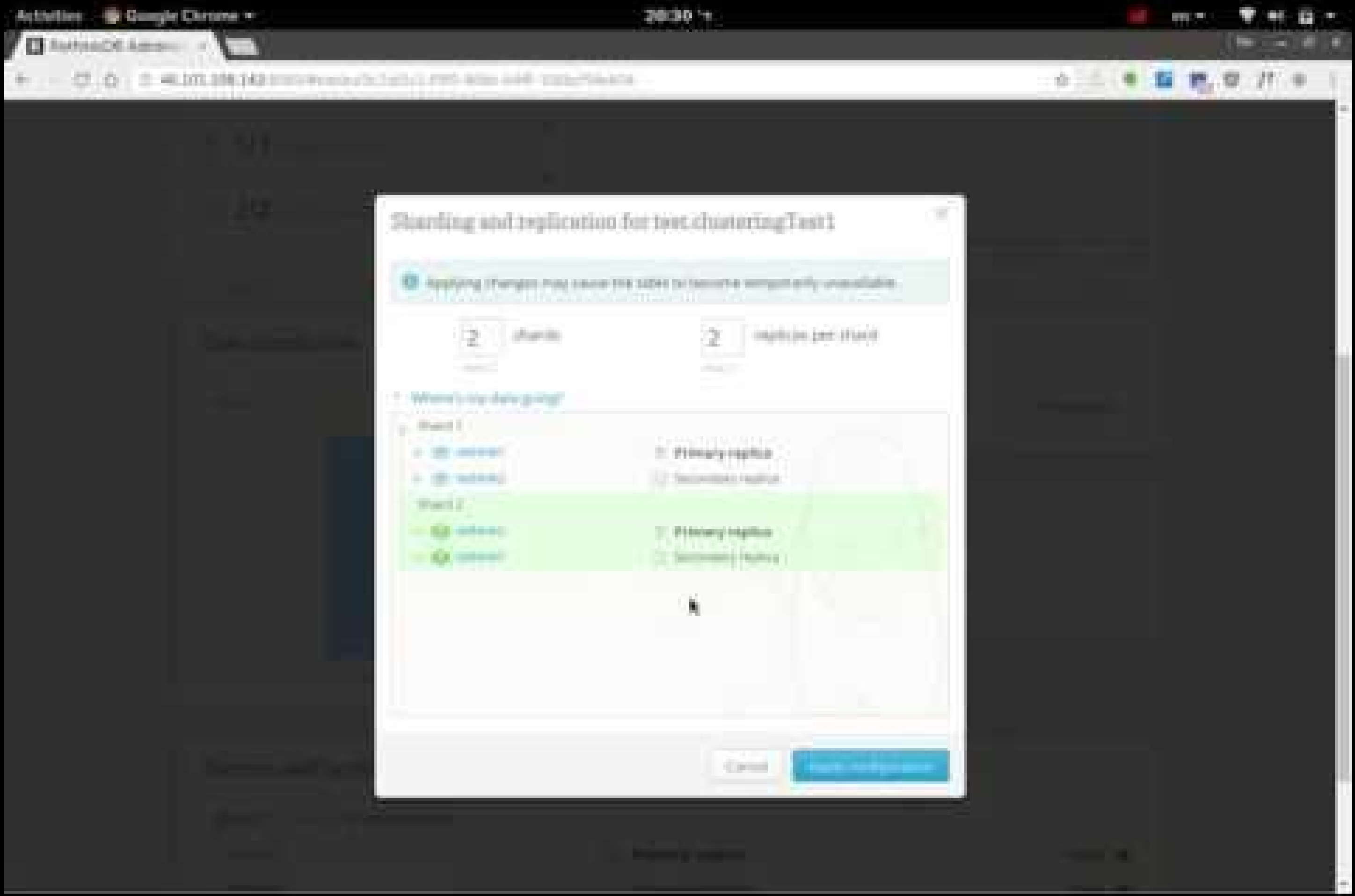
```
➜ ~ ssh root@85.85.6.109
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.


Last login: Fri Aug  5 08:29:48 2016 from 79.180.97.188
root@remoteub-2:~# 
```

2. Scalability - Sharding

Sharding is a distribution of the table to number of machines. This allows us to store more information, and handle more traffic without vertical scalability - buy a more powerful machine.

What we’ll do:

- Create a sharding to our ‘clusteringTest1’ table.



Sharding and replication for test.clusteringTest1

Applying changes may cause the table to become temporarily unavailable.

2 shards

2 replicas per shard

Where's my data going?

Shard 1

[illegible] Primary replica

[illegible] Secondary replica

Shard 2

[illegible] Primary replica

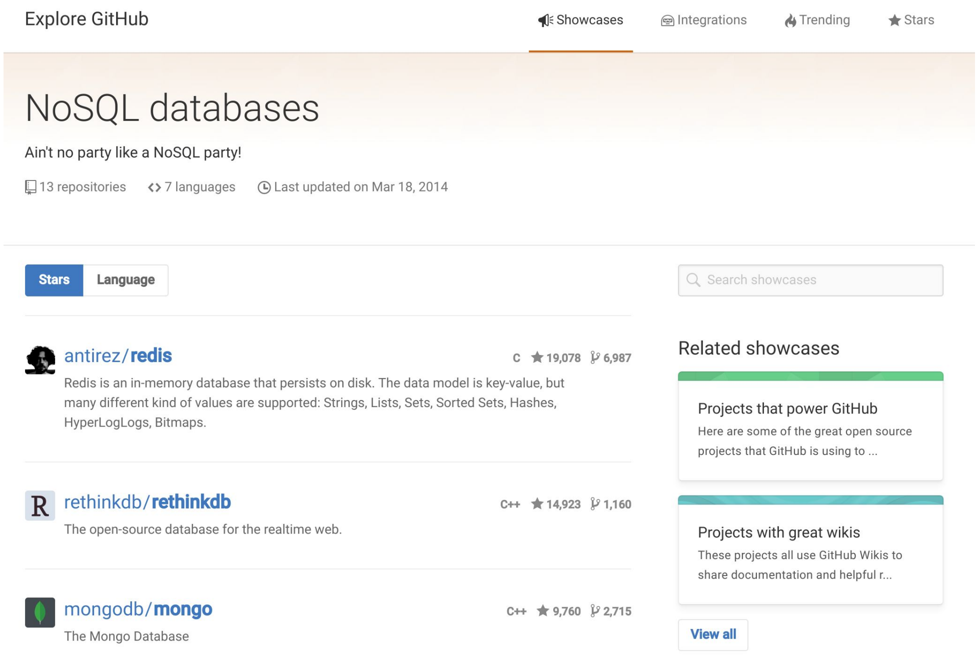
[illegible] Secondary replica

3. Open, large, and vibrant community

RethinkDB is the second most popular NoSQL database in GitHub [link].

They send us goodies to this meetup.

They active on slack, twitter, IRC, Stack Overflow, Google groups, their GitHub repo [link].



NoSQL databases

Ain't no party like a NoSQL party!

13 repositories · 7 languages · Last updated on Mar 18, 2014

antirez/redis — C ★ 19,078 6,987

Redis is an in-memory database that persists on disk. The data model is key-value, but many different kind of values are supported: Strings, Lists, Sets, Sorted Sets, Hashes, HyperLogLogs, Bitmaps.

rethinkdb/rethinkdb — C++ ★ 14,923 1,160

The open-source database for the realtime web.

mongodb/mongo — C++ ★ 9,760 2,715

The Mongo Database

Related showcases

Projects that power GitHub

Here are some of the great open source projects that GitHub is using to ...

Projects with great wikis

These projects all use GitHub Wikis to share documentation and helpful r...

4. IoT DB (or Lock-free architecture)

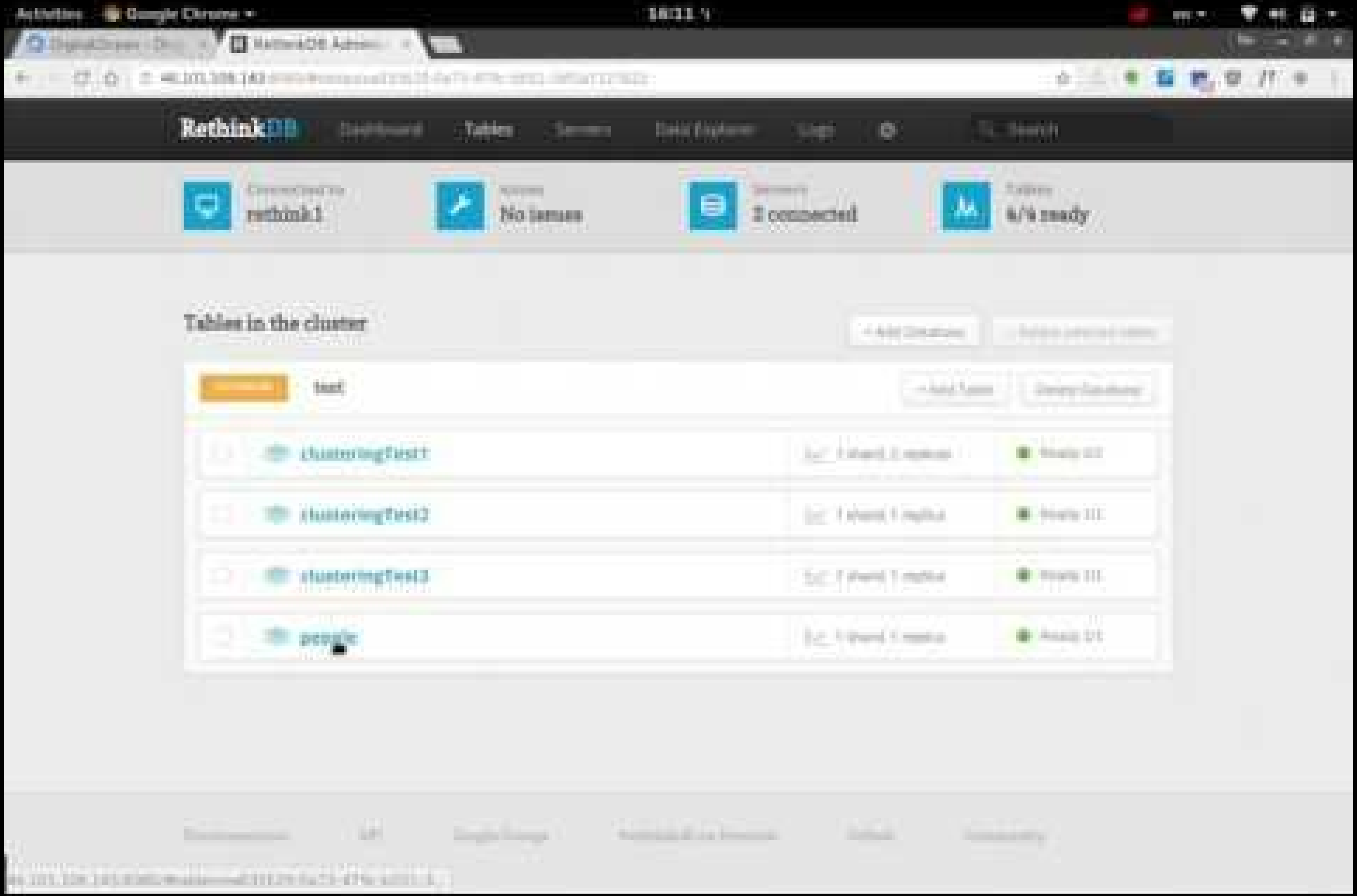
RethinkDB uses MVCC (Multi-Version Concurrency Control). In short it's means reading **never** blocks writing and vice versa.

Whenever a write operation occurs while there is an ongoing read, RethinkDB takes a snapshot of the B-Tree for each relevant shard and temporarily maintains different versions of the blocks in order to execute read and write operations concurrently. From the perspective of the applications written on top of RethinkDB, the system is essentially lock-free— you can run an hour-long analytics query on a live system without blocking any real-time reads or writes.

5. Performance tuning

RethinkDB support various types of indexes:

- Simple indexes - constructed on the value of a single field within a document
- Compound indexes - based on multiple fields.



Connected to rethink1

Issues No issues

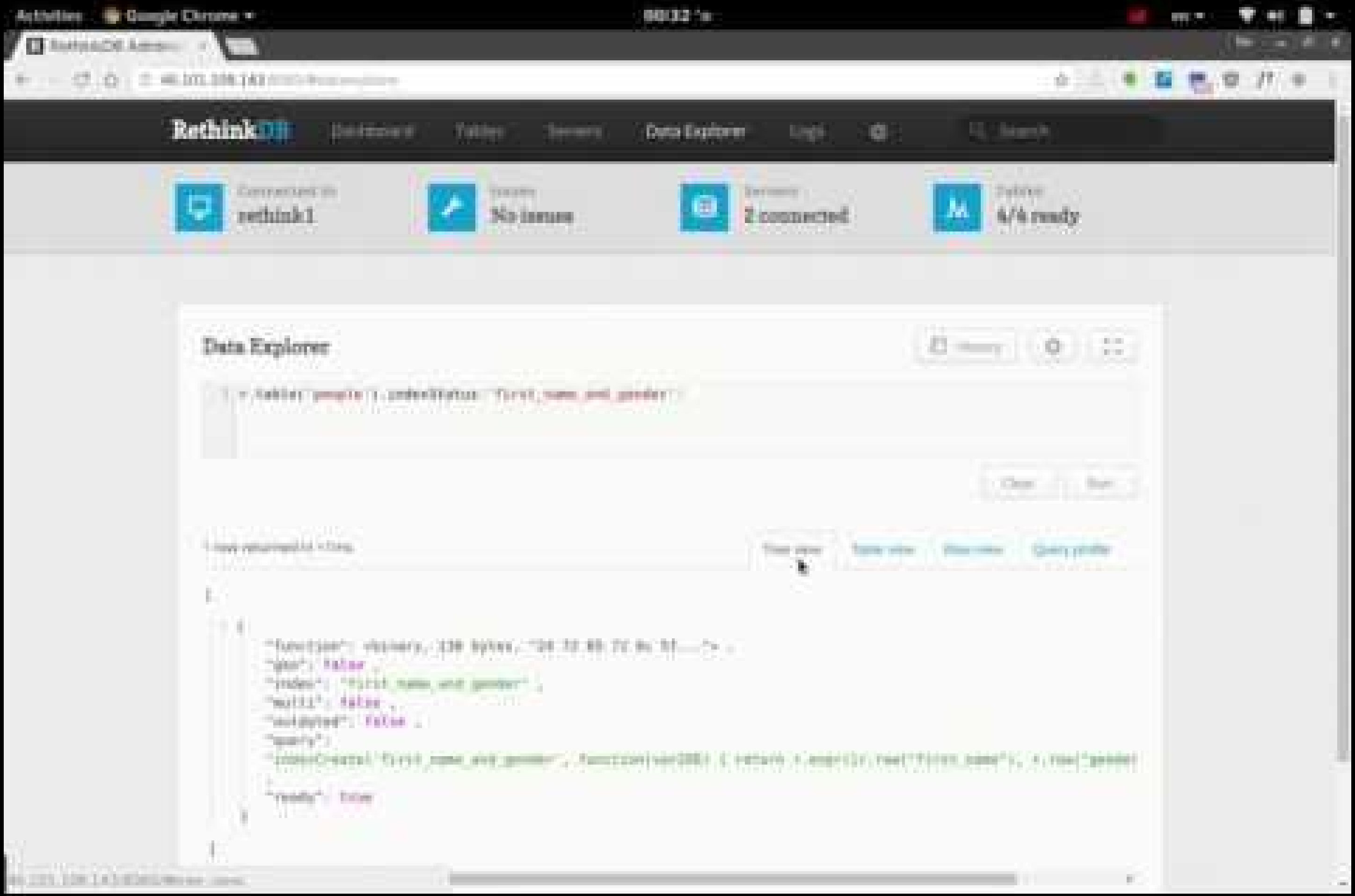
Servers 2 connected

Tables 4/4 ready

Tables in the cluster

test

clusteringTest1	1 shard, 2 replicas	Ready 2/2
clusteringTest2	1 shard, 1 replica	Ready 1/1
clusteringTest3	1 shard, 1 replica	Ready 1/1
people	1 shard, 1 replica	Ready 1/1



```
r.table('people').indexStatus('first_name_and_gender')
```

1 row returned in [illegible]

```
[
  {
    "function": <binary, 130 bytes, "24 72 65 72 0a 51...">,
    "geo": false ,
    "index": "first_name_and_gender" ,
    "multi": false ,
    "outdated": false ,
    "query":
    "indexCreate('first_name_and_gender', function(var205) { return r.expr([r.row("first_name"), r.row("gende
    ,
    "ready": true
  }
]
```

6. Distributed joins

RethinkDB not only supports joins but automatically compiles them to distributed programs and executes them across the cluster without further intervention from the client.

- Inner Join.
- Outer Join.
- Equal Join.

Example: r.table('people').eqJoin('username', r.table('orders'))

Where do you go from here ?

Go to https://rethinkdb.com/

And just build your next app with RethinkDB !

Breaktime !