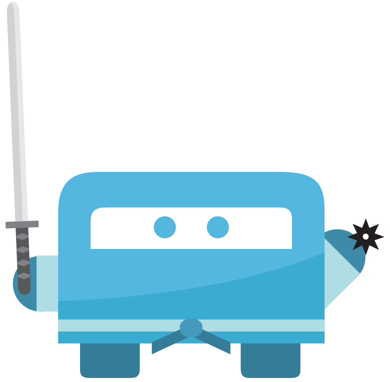
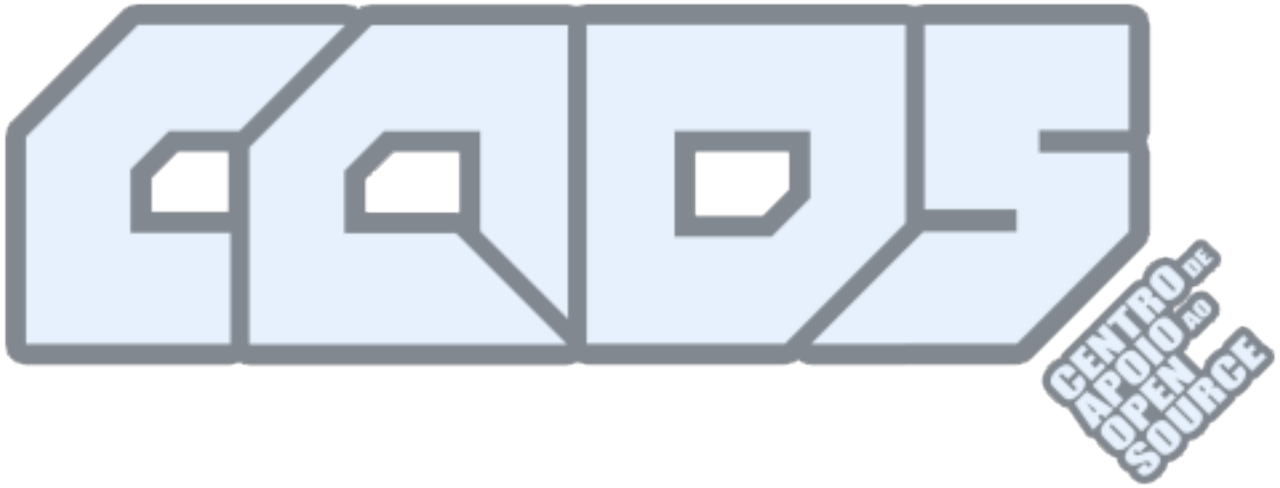
Rocking show-case to get you excited about

pictures of my cats relaxing in the only remaining piece of furniture yet to be used to name a CouchDB project: bean bag!

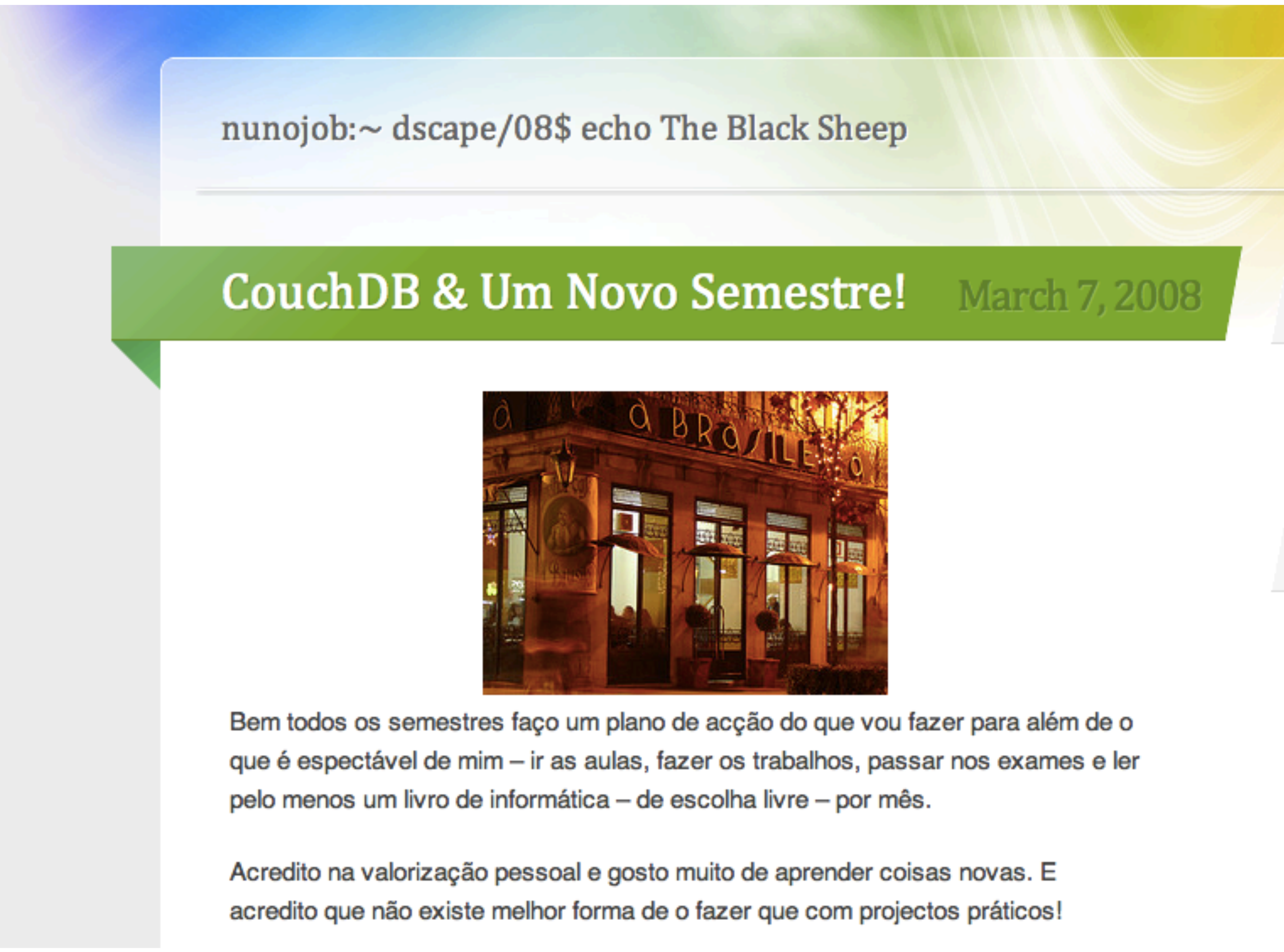
ohai!



```
$ whoami
@dscape
```


CAOS
CENTRO DE
APOIO AO
OPEN
SOURCE

my ambition is school was to hack on CouchDB



nunojob:~ dscape/08$ echo The Black Sheep

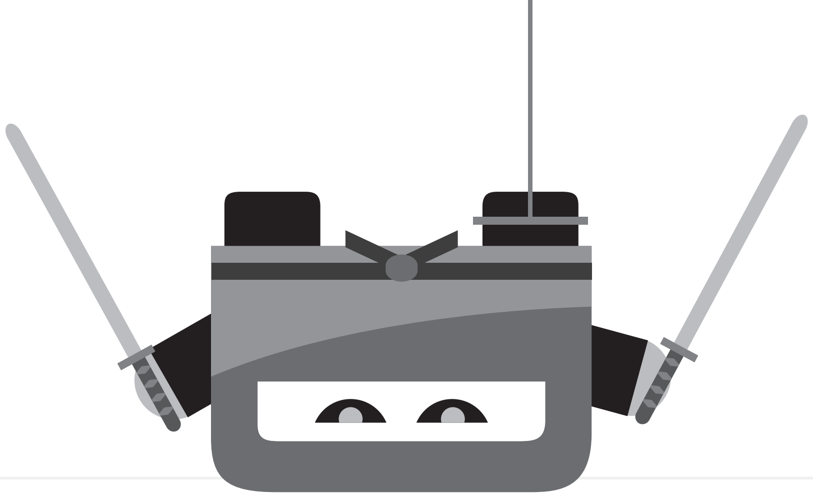
CouchDB & Um Novo Semestre!

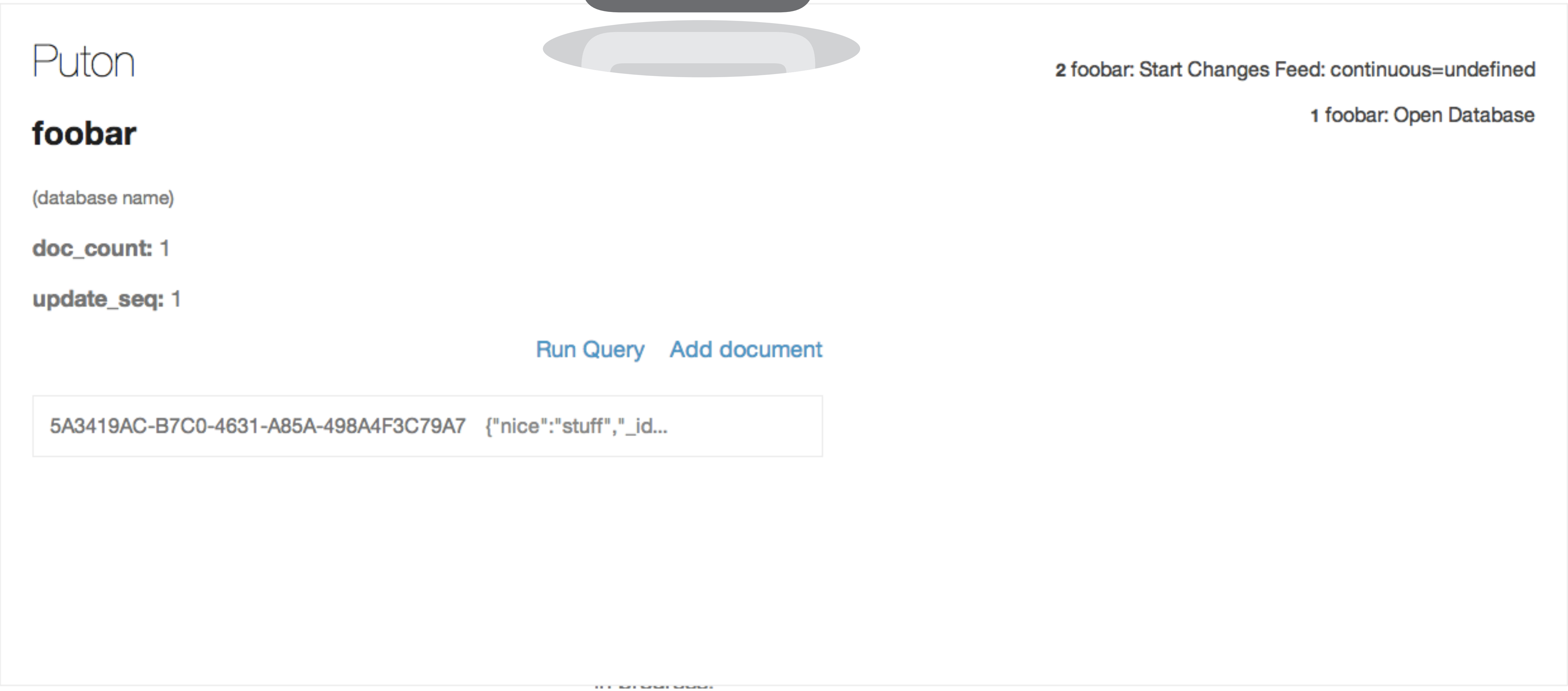
March 7, 2008

Bem todos os semestres faço um plano de acção do que vou fazer para além de o que é espectável de mim – ir as aulas, fazer os trabalhos, passar nos exames e ler pelo menos um livro de informática – de escolha livre – por mês.

Acredito na valorização pessoal e gosto muito de aprender coisas novas. E acredito que não existe melhor forma de o fazer que com projectos práticos!

SONY
VAIO
nodejitsu
BLE





Puton

foobar

(database name)

doc_count: 1

update_seq: 1

```
5A3419AC-B7C0-4631-A85A-498A4F3C79A7   {"nice":"stuff","_id...
```

```
2 foobar: Start Changes Feed: continuous=undefined
1 foobar: Open Database
```

```
var db_name = 'hackaton_' + (~~(Math.random() * 1e9)).toString(36);

var pouch;

Pouch('idb://'+db_name, function (err, db) {
  pouch=db;
});

pouch.put({_id: '1', bar: 'baz'});

pouch.get('1', function (err, doc) { console.log(doc); });

Pouch.replicate('idb://' + db_name,
'http://localhost:2020/foobar2003', function(err, changes) {
  console.log('omg hi!')
  console.log('got changes', changes);
});
```

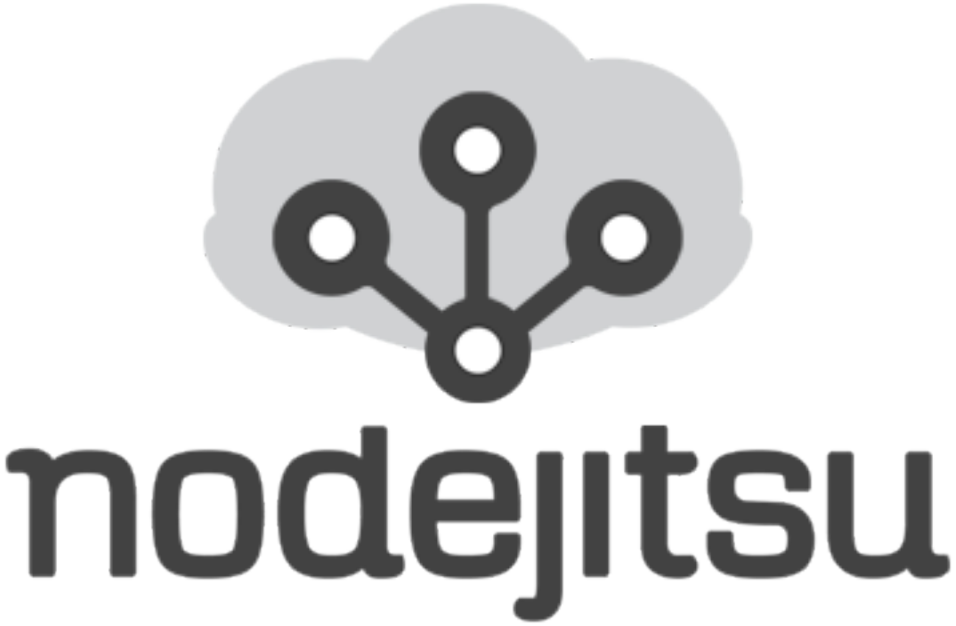
github.com/daleharvey/pouchdb

!!111!!!1!!one!!1!

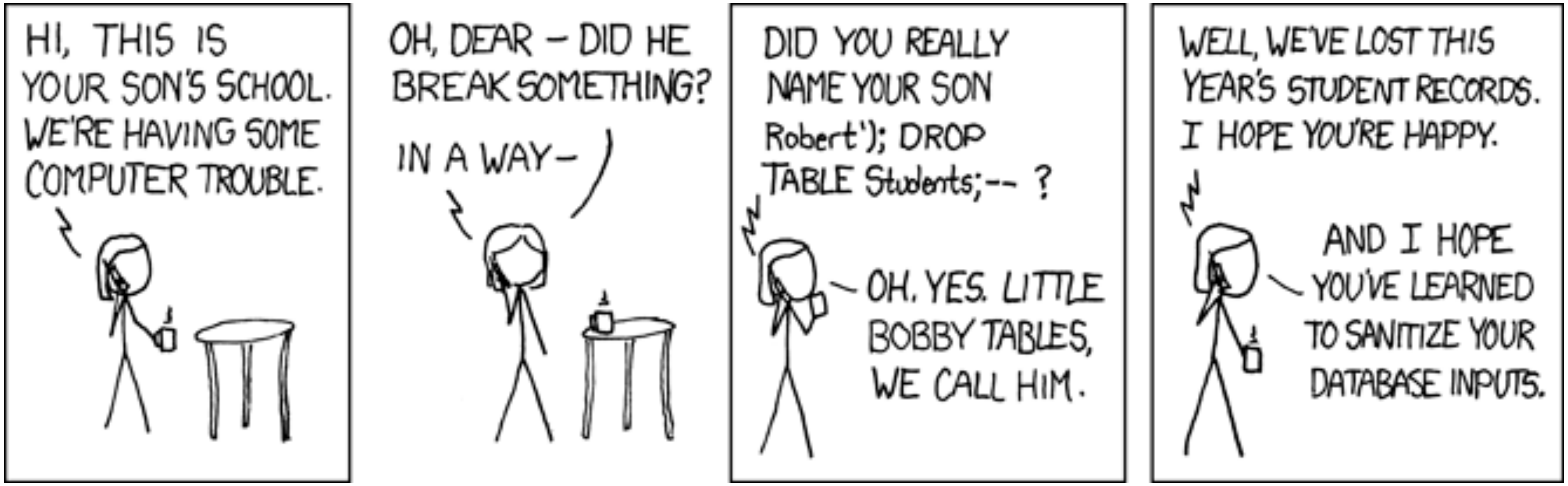
```
var db_name = 'hackaton_' + (~~(Math.random() * 1e9)).toString(36);

var pouch;

Pouch('idb://'+db_name, function (err, db) {
  pouch=db;
});

pouch.put({_id: '1', bar: 'baz'});

pouch.get('1', function (err, doc) { console.log(doc); });

Pouch.replicate('idb://' + db_name,
'http://localhost:2020/foobar2003', function(err, changes) {
  console.log('omg hi!');
  console.log('got changes', changes);
});
```

http://git.io/IxpQRg

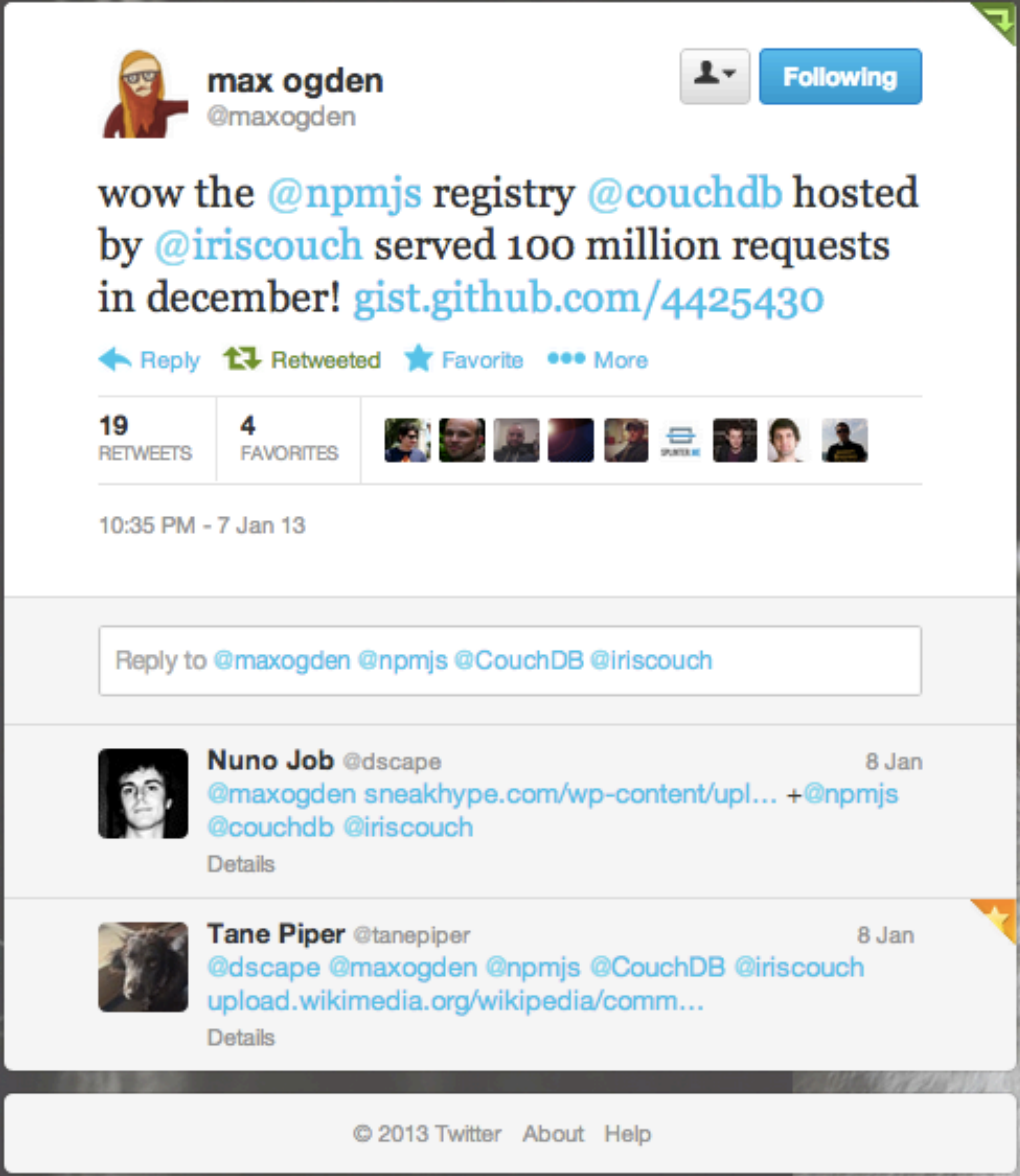
NODE
FIRM


nodejitsu

HELLO
YES, THIS IS DOG


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

couchdb <3’s nodejs
nodejs <3’s couchdb



max ogden @maxogden

wow the @npmjs registry @couchdb hosted by @iriscouch served 100 million requests in december! gist.github.com/4425430

19 RETWEETS

4 FAVORITES

10:35 PM - 7 Jan 13

Reply to @maxogden @npmjs @CouchDB @iriscouch

Nuno Job @dscape 8 Jan
@maxogden sneakhype.com/wp-content/upl... +@npmjs @couchdb @iriscouch
Details

Tane Piper @tanepiper 8 Jan
@dscape @maxogden @npmjs @CouchDB @iriscouch upload.wikimedia.org/wikipedia/comm...
Details

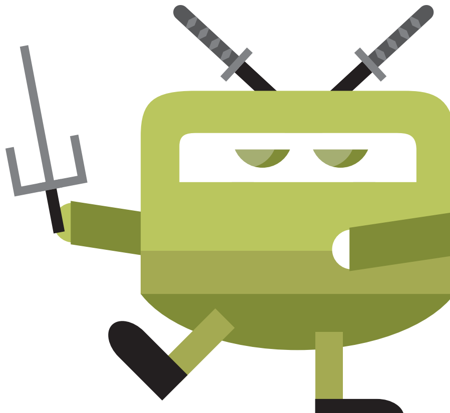
© 2013 Twitter About Help

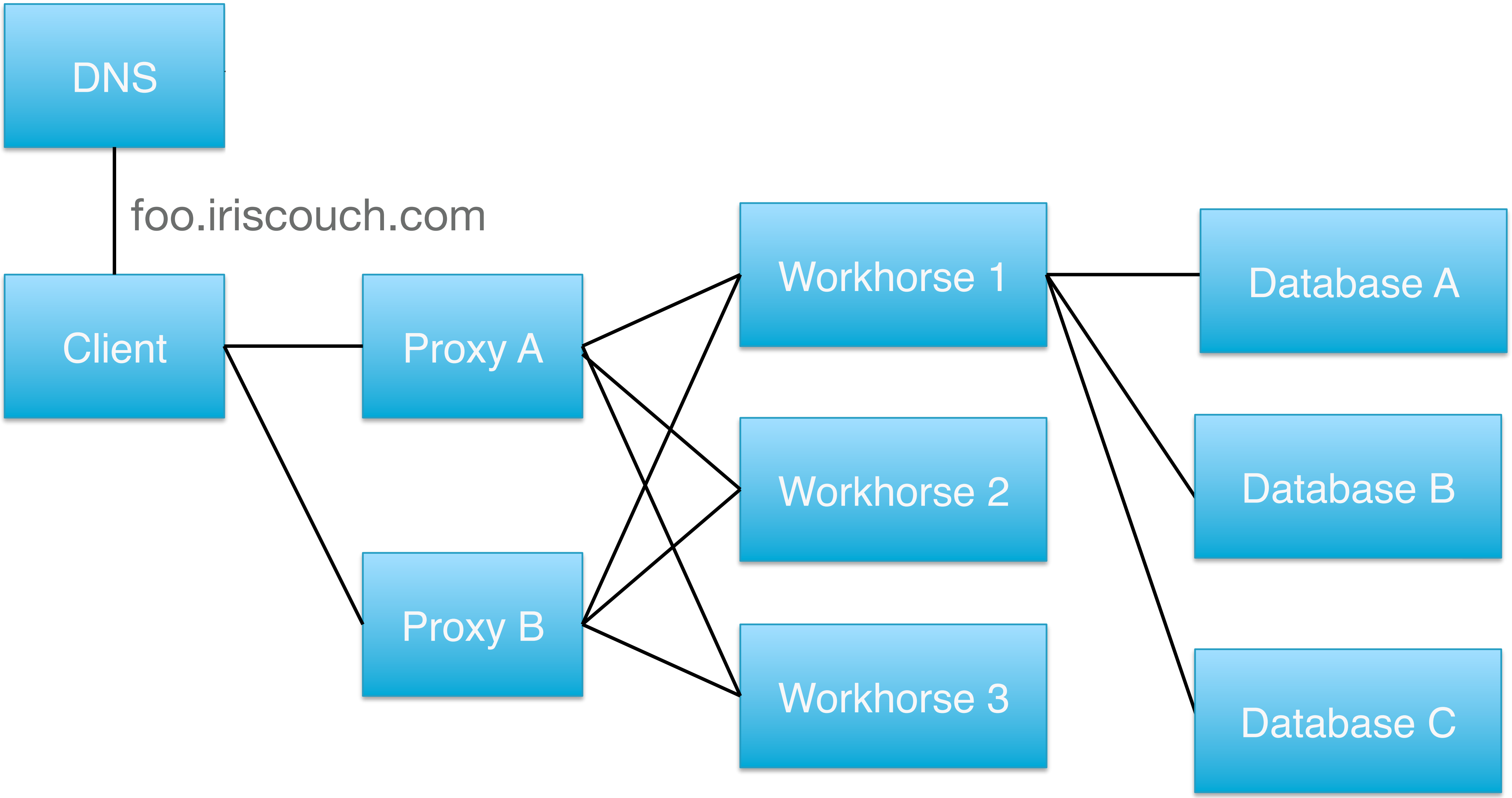
peek: ~1k req/s on a package repository

npm

Iris Couch

how does it work?




DNS
foo.iriscouch.com
Client
Proxy A
Proxy B
Workhorse 1
Workhorse 2
Workhorse 3
Database A
Database B
Database C

FLIGHT
RECORDER
DO NOT
OPEN

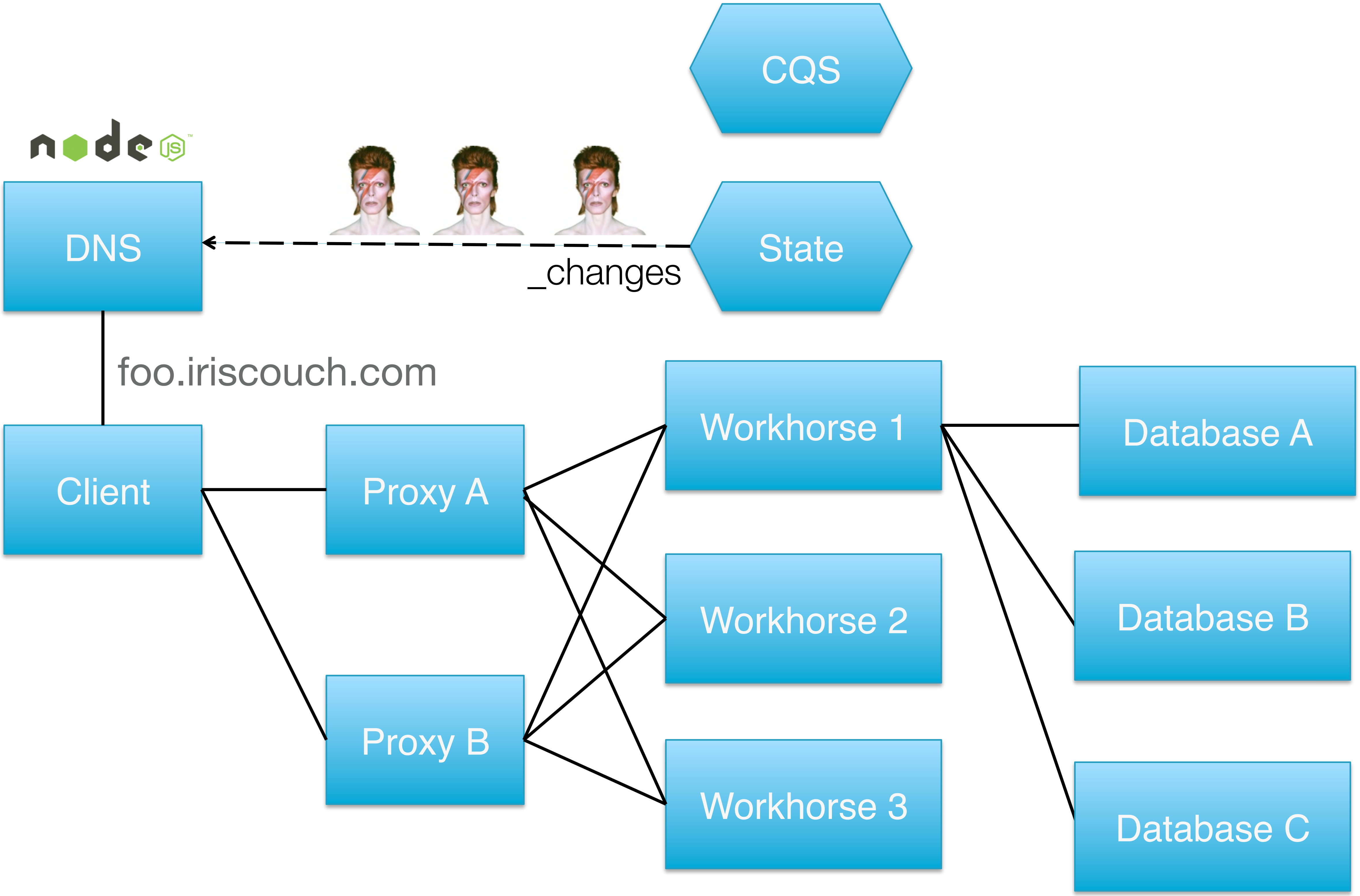
```
var named = require('named');
named.createServer(function(req, res) {
  res.end('1.2.3.4');
}).listen(5353, '127.0.0.1');
console.log('Server running')
```

github.com/iriscouch/named

again!11!!!11!

```
var named = require('named');
named.createServer(function(req, res) {
  res.end('1.2.3.4');
}).listen(5353, '127.0.0.1');
console.log('Server running');
```

github.com/iriscouch/named



CQS

DNS

State

_changes

foo.iriscouch.com

Client

Proxy A

Proxy B

Workhorse 1

Workhorse 2

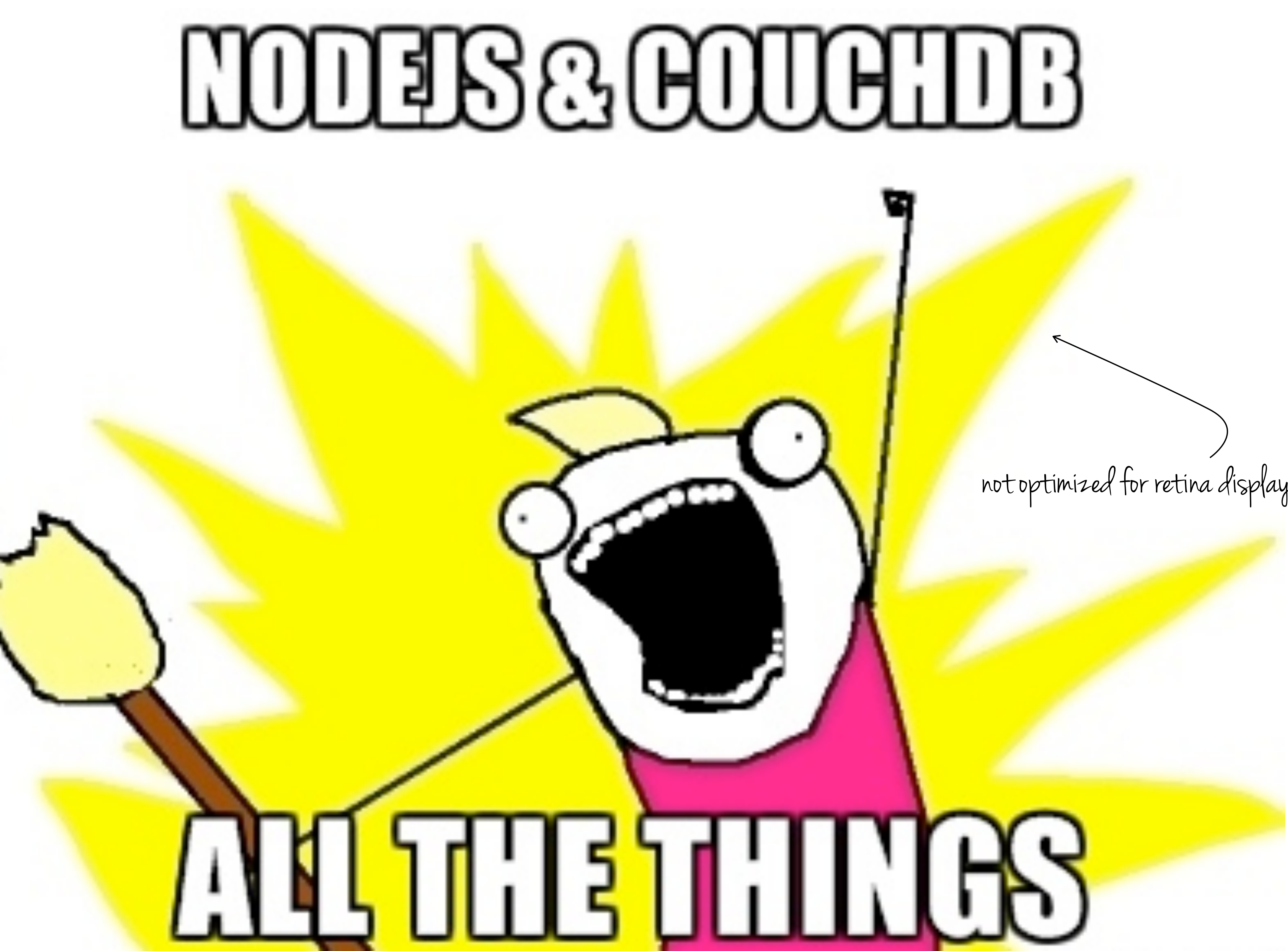
Workhorse 3

Database A

Database B

Database C

the nodejitsu way



not optimized for retina displays

open source tools we made/use to help us manage our couches

nano.relax(opts, [callback])

alias for `nano.request`

nano.dinosaur(opts, [callback])

alias for `nano.request`

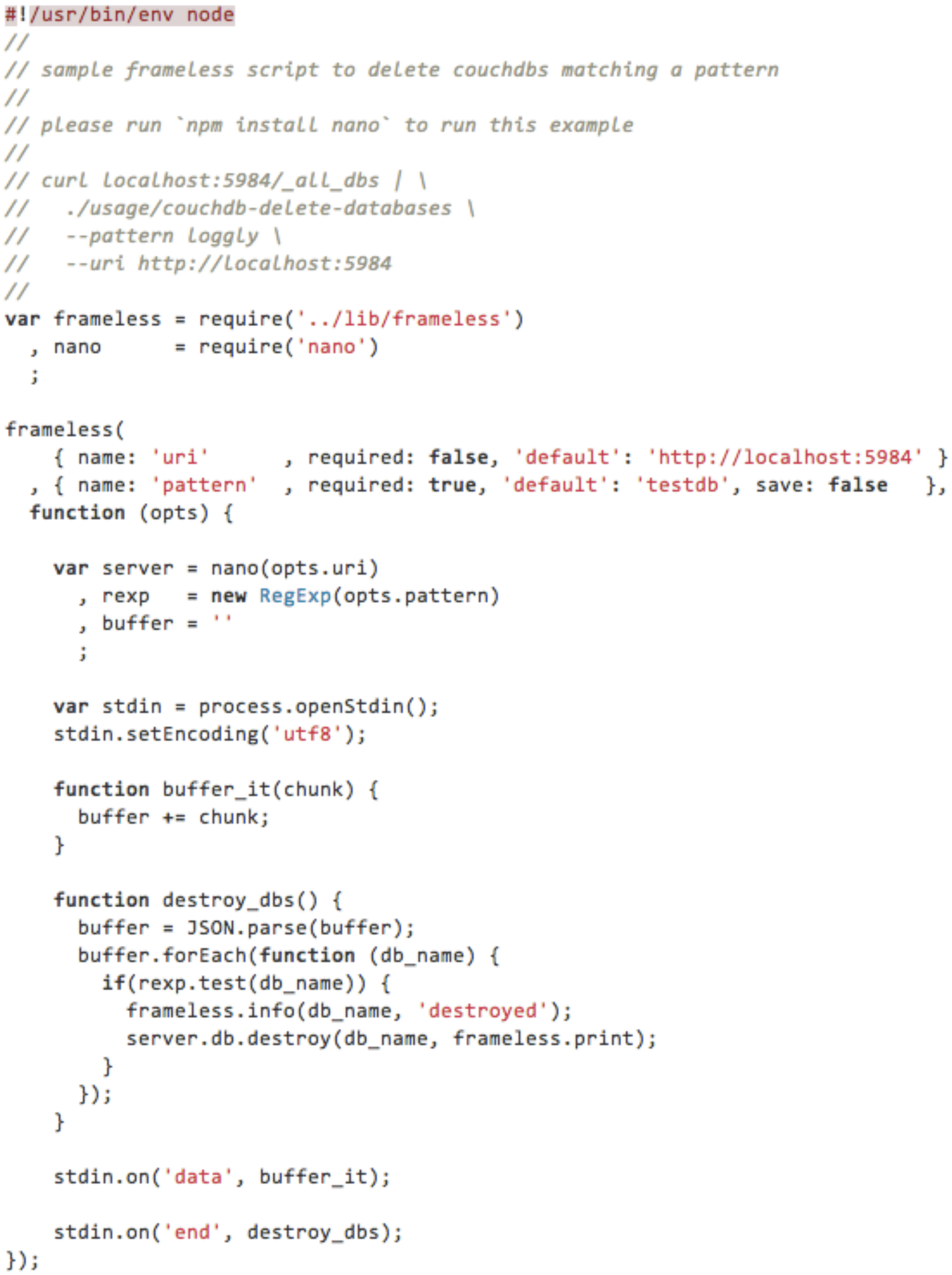
```
               _
             / '_)  WAT U SAY!
    _.----._/  /
   /          /
 _/  (   | ( |
/__.-|_|--|_l
```

nano.config

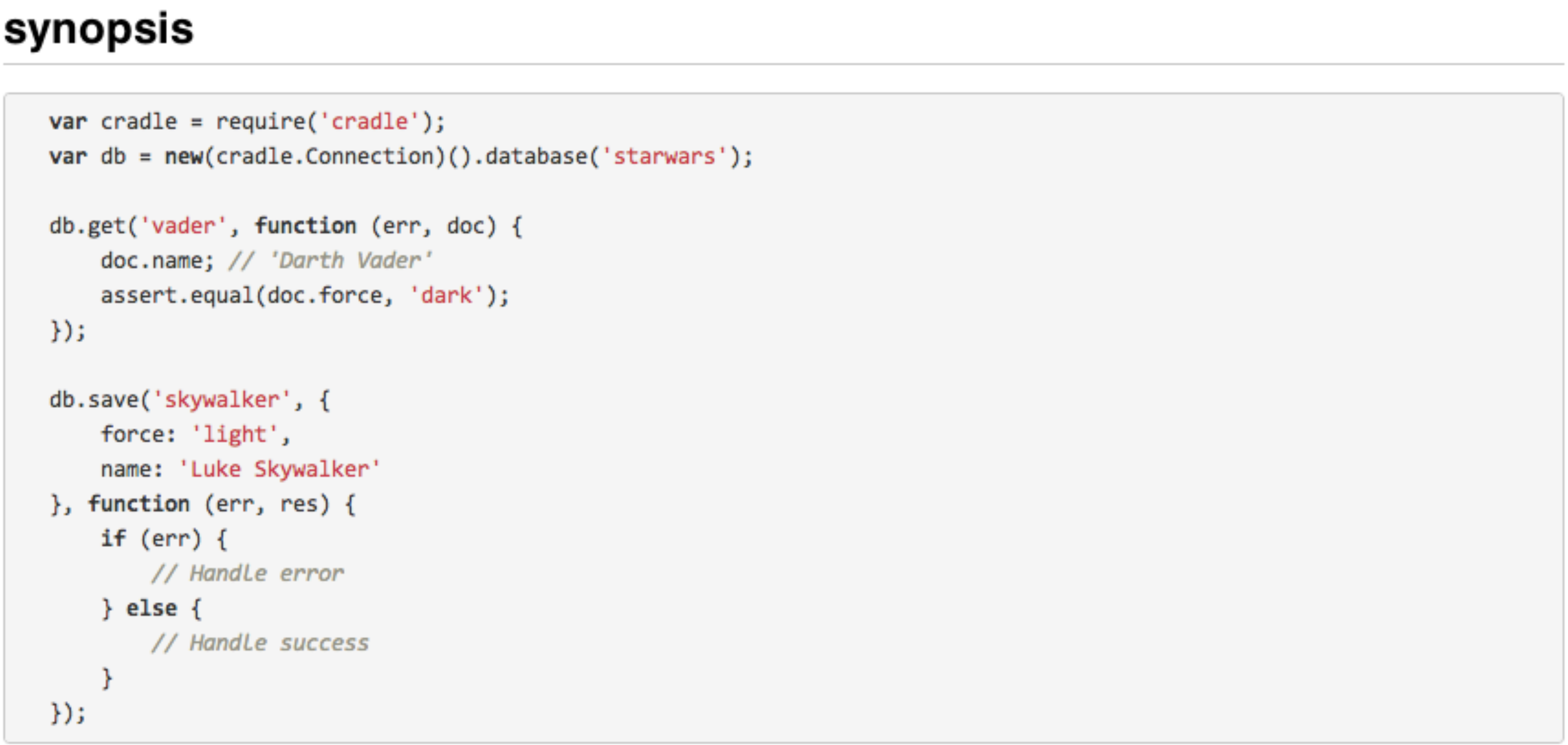
an object containing the nano configurations, possible keys are:

- `url` - the couchdb url
- `db` - the database name

github.com/dscape/nano



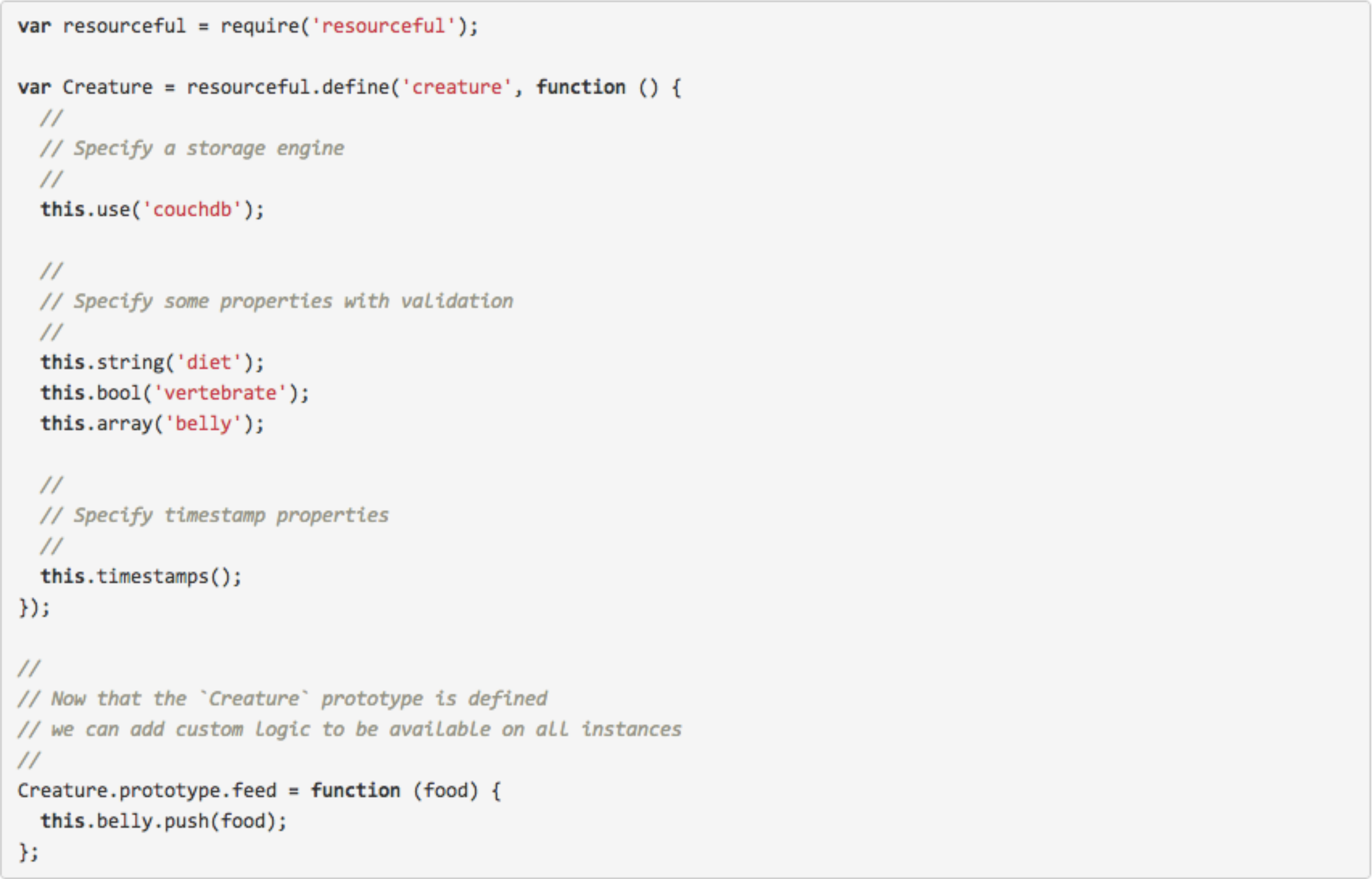
```
#!/usr/bin/env node
//
// sample frameless script to delete couchdbs matching a pattern
//
// please run `npm install nano` to run this example
//
// curl localhost:5984/_all_dbs | \
//   ./usage/couchdb-delete-databases \
//   --pattern loggly \
//   --uri http://localhost:5984
//
var frameless = require('../lib/frameless')
  , nano      = require('nano')
  ;

frameless(
    { name: 'uri'      , required: false, 'default': 'http://localhost:5984' }
  , { name: 'pattern'  , required: true, 'default': 'testdb', save: false   },
  function (opts) {

    var server = nano(opts.uri)
      , rexp   = new RegExp(opts.pattern)
      , buffer = ''
      ;

    var stdin = process.openStdin();
    stdin.setEncoding('utf8');

    function buffer_it(chunk) {
      buffer += chunk;
    }

    function destroy_dbs() {
      buffer = JSON.parse(buffer);
      buffer.forEach(function (db_name) {
        if(rexp.test(db_name)) {
          frameless.info(db_name, 'destroyed');
          server.db.destroy(db_name, frameless.print);
        }
      });
    }

    stdin.on('data', buffer_it);

    stdin.on('end', destroy_dbs);
});
```



synopsis

```
var cradle = require('cradle');
var db = new(cradle.Connection)().database('starwars');

db.get('vader', function (err, doc) {
    doc.name; // 'Darth Vader'
    assert.equal(doc.force, 'dark');
});

db.save('skywalker', {
    force: 'light',
    name: 'Luke Skywalker'
}, function (err, res) {
    if (err) {
        // Handle error
    } else {
        // Handle success
    }
});
```

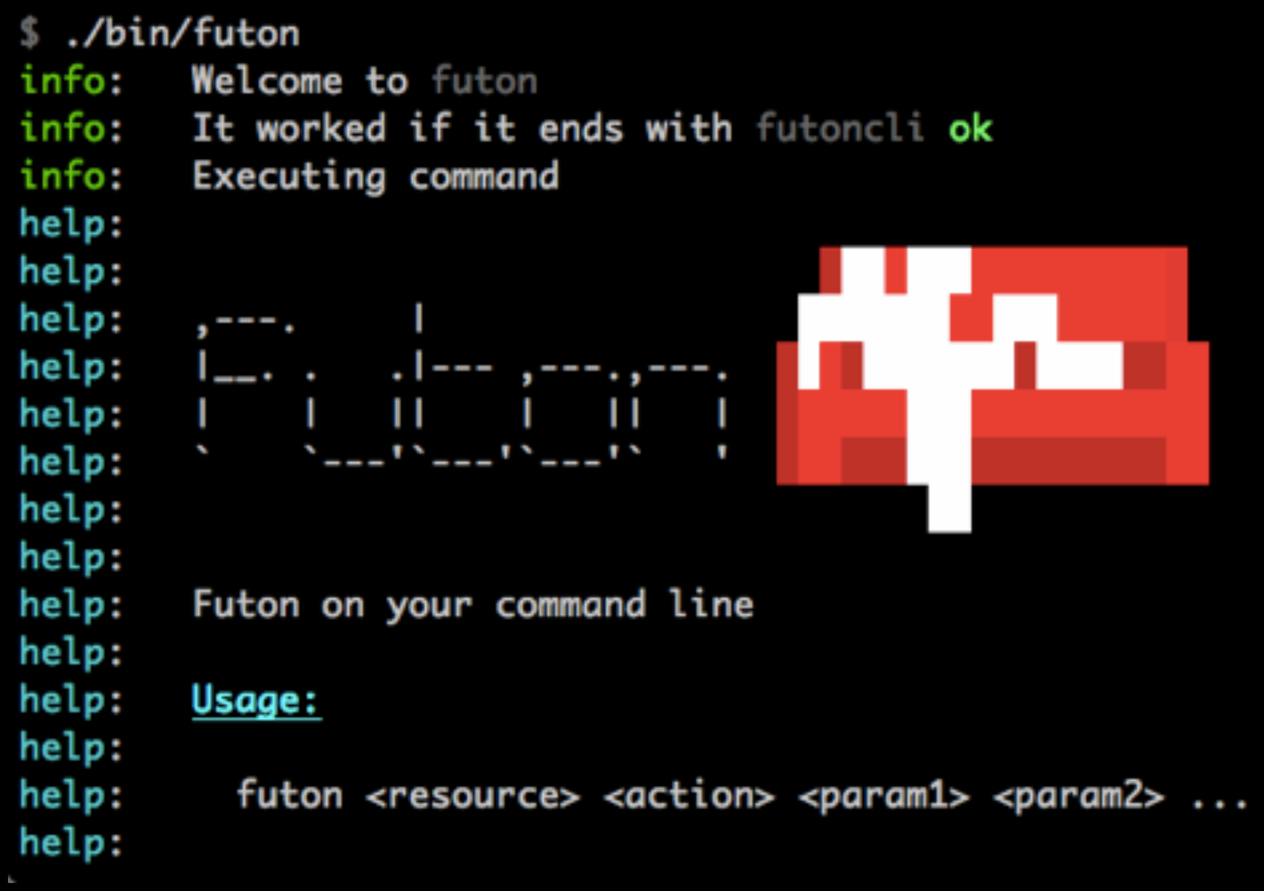
github.com/cloudhead/cradle



```
var resourceful = require('resourceful');

var Creature = resourceful.define('creature', function () {
  //
  // Specify a storage engine
  //
  this.use('couchdb');

  //
  // Specify some properties with validation
  //
  this.string('diet');
  this.bool('vertebrate');
  this.array('belly');

  //
  // Specify timestamp properties
  //
  this.timestamps();
});

//
// Now that the `Creature` prototype is defined
// we can add custom logic to be available on all instances
//
Creature.prototype.feed = function (food) {
  this.belly.push(food);
};
```

github.com/nodejitsu/resourceful

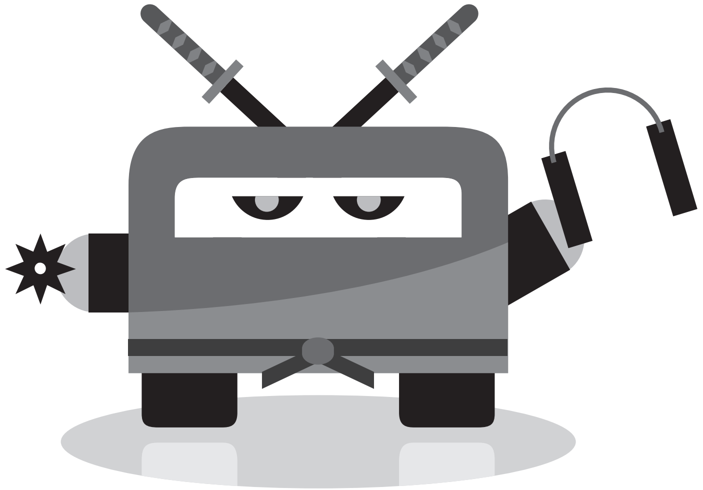
npm install -g futon



```
$ ./bin/futon
info:   Welcome to futon
info:   It worked if it ends with futoncli ok
info:   Executing command
help:
help:
help:   ,---.     |
help:   |__. .   .|--- ,---.,---.
help:   |    |   ||    |   ||   |
help:   `    `---'`---'`---'`   '
help:
help:
help:   Futon on your command line
help:
help:   Usage:
help:
help:     futon <resource> <action> <param1> <param2> ...
help:
```

github.com/dscape/futoncli

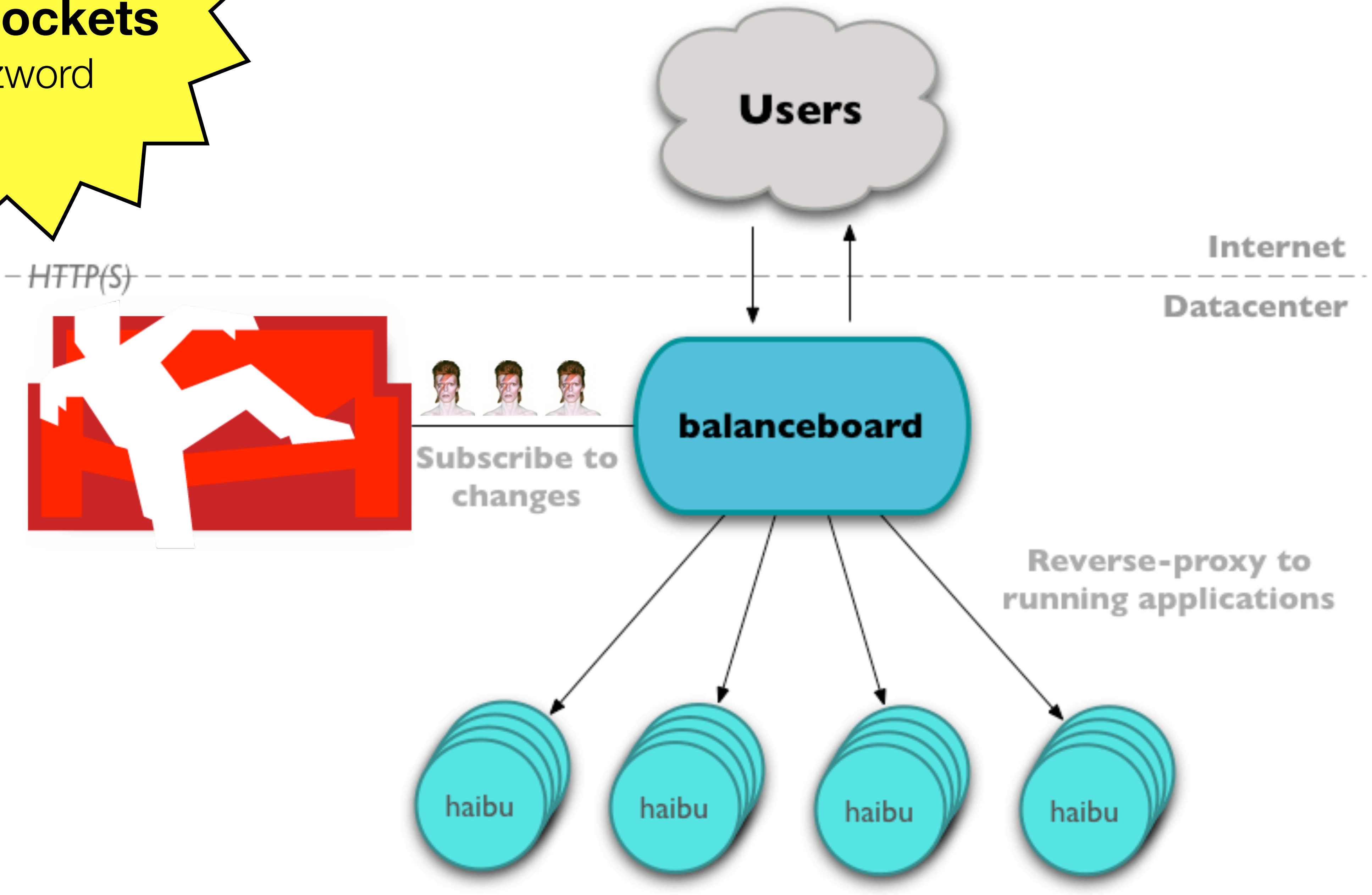
how to use CouchDB to build your own open source



load balancers

reverse-proxy fully compatible with **web-sockets** buzzword

yes, sticky sessions too!



Users

Internet

Datacenter

HTTP(S)

balanceboard

Subscribe to changes

Reverse-proxy to running applications

haibu

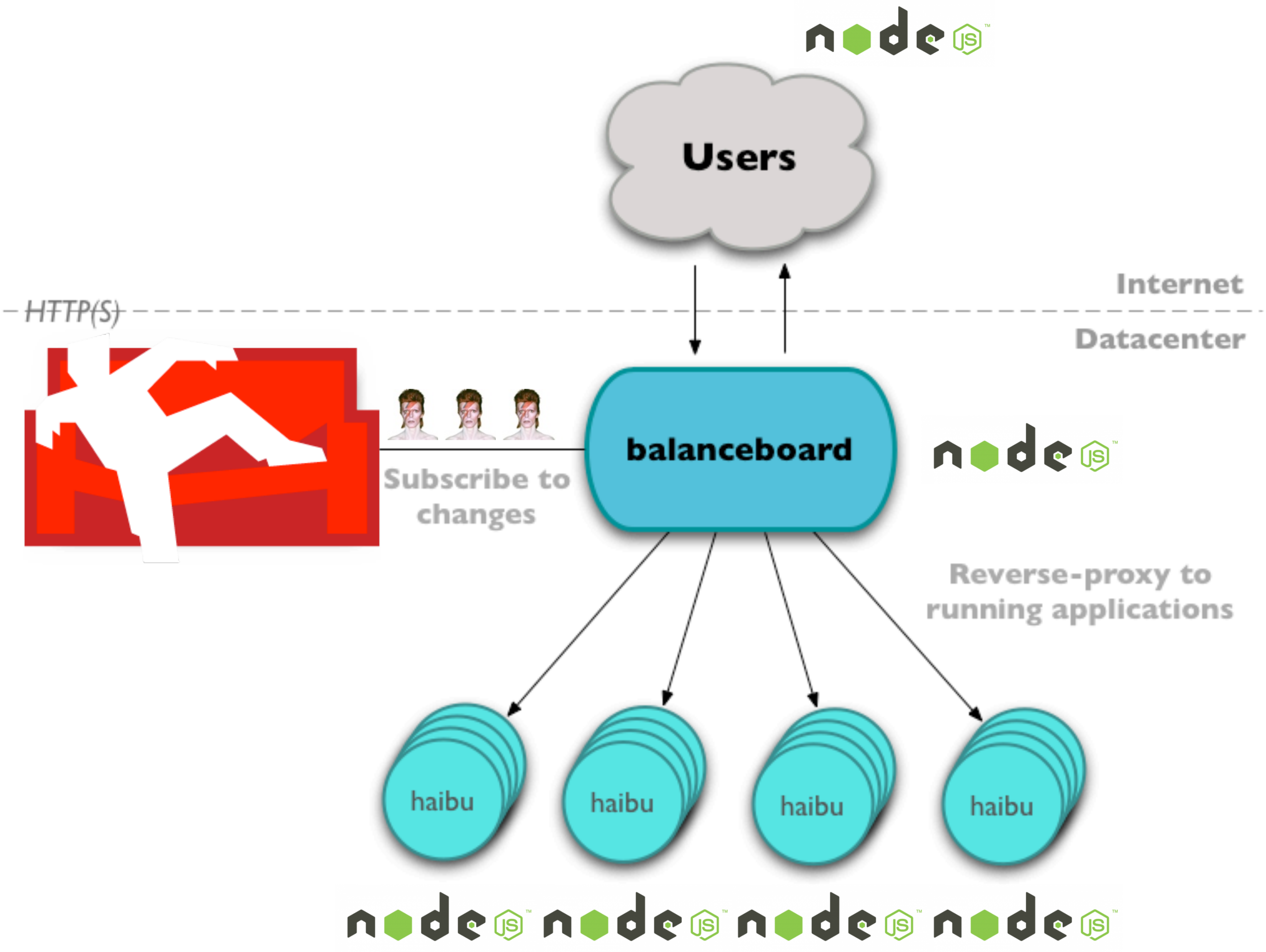
haibu

haibu

haibu

github.com/nodejitsu/{haibu, node-http-proxy}

wondering about the technology stack?

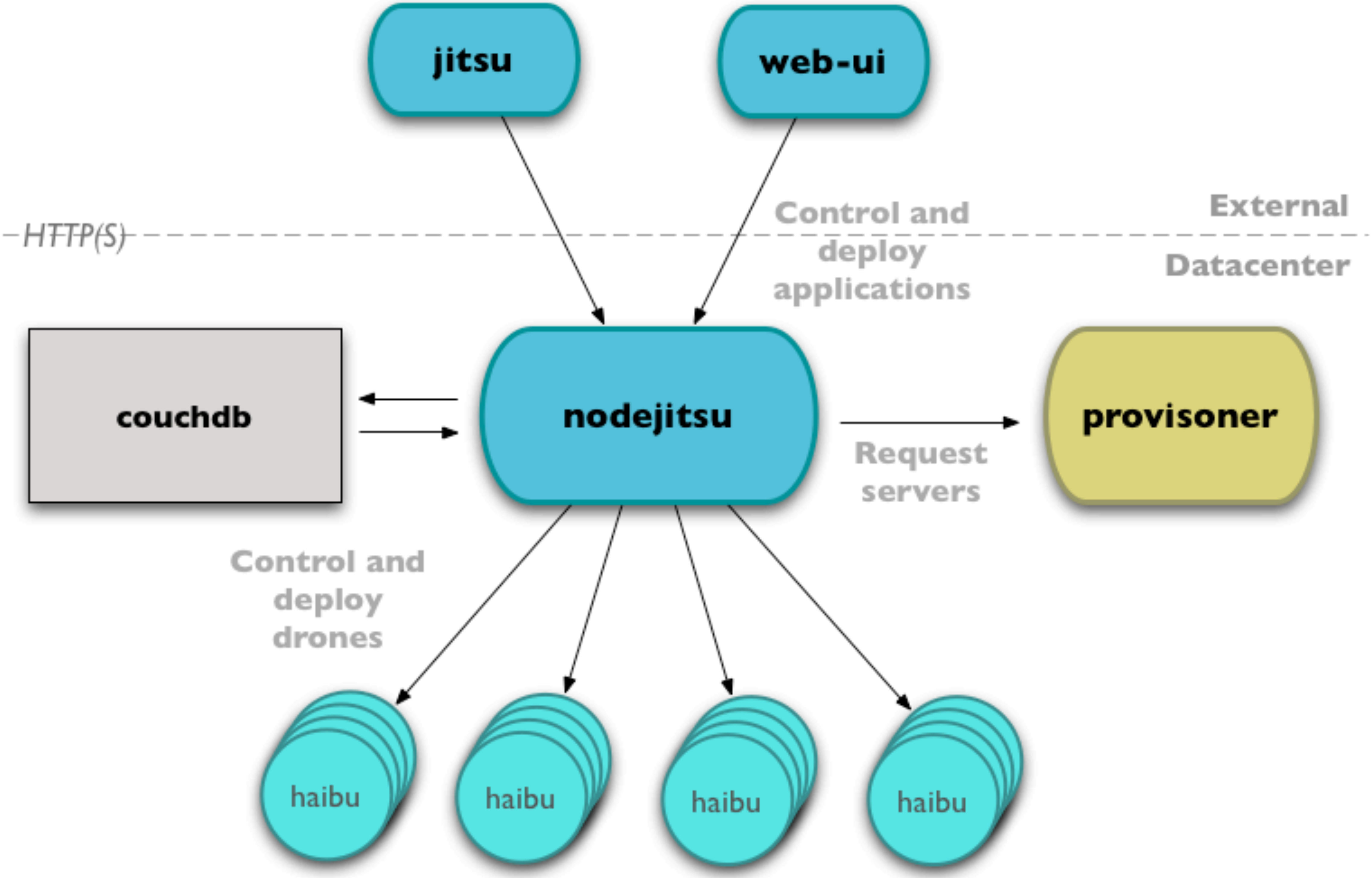


github.com/nodejitsu/{haibu, node-http-proxy}

dogmatic?

we are fast at prototyping with node and built a team around that particular skill whenever requirements change beyond what node can offer we translate our stack to different frameworks/languages but yes, maybe... a little!

deployment



jitsu

web-ui

External

HTTP(S)

Control and deploy applications

Datacenter

couchdb

nodejitsu

provisoner

Request servers

Control and deploy drones

haibu

haibu

haibu

haibu

deep dive

fun right?

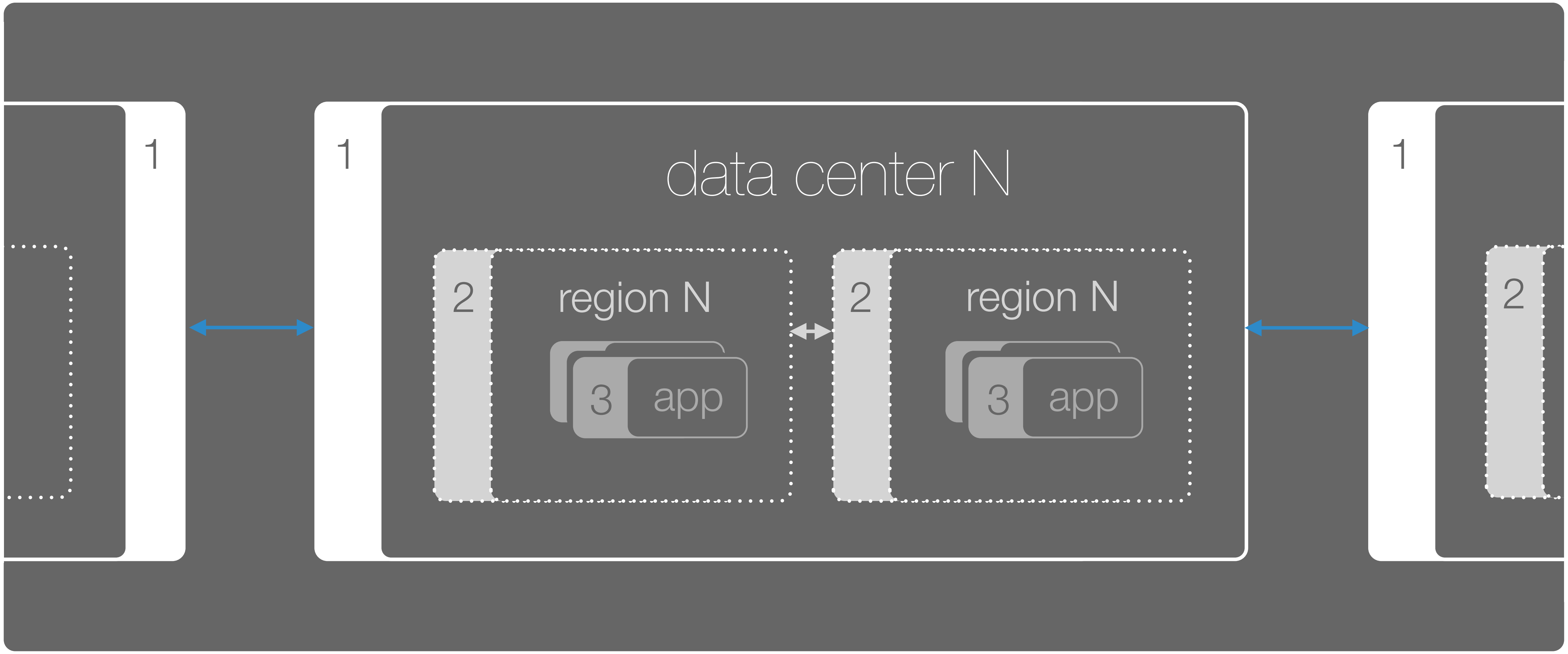
migrating to CouchDB 1.2.x next week

tips are welcome!

stud for 6984

_changes feed to provide realtime deployment status for our continuous deployments private replica of the public npm registry provisioner stores server pool in CouchDB private npm for deploying our systems

e.g. continuous deployments or our load balancers coming soon



github.com/nodejitsu/pkgcloud

Dankie Shnorhakalutiun Hvala Blagodariya Hvala Dekuju Tak Kiitos Merci Grazie Grazzi Takk Danke Thank You Dziękuję Obrigado Gracias Asante Tack Teshekur Ederim Shokriya Cam On Kob Kun Kaa Mamnun Arigatou