```
$ curl -XPUT "http://localhost:8098/riak/presentations/riak" \
--header "Content-Type: application/json" \
--data '{
         "subject": "Riak",
         "presenters": [
           ["Sam Stelfox", "@SamStelfox", "http://stelfox.net" ],
           ["Gabe Koss",   "@granolocks", "http://gabekoss.com"]
         ]
       }'
```

“NoSQL marketing is confusing... Everything does everything and at a small scale everything works.”

“If you're evaluating Mongo vs. Riak or Couch vs. Cassandra you don't understand either your problem or the technologies.”

- Andy Gross, VP Engineering at Basho Technologies

(a*pproximate paraphrasing, hopefully not grossly misquoted*)

# What does Riak give me?

- **HTTP access**
- **Simple Key-Value Database**
- **Masterless Clustering**
- **CAP Tuning**
- **Map/Reduce using Javascript or Erlang**
- Full Text Search via Solr
- Linear Scalability
- Link Walking
- Automatic Sharding / Consistent Hashing
- Commit Hooks using Javascript or Erlang
- Secondary Indexes
- “Shared-nothing” Architecture

# Basically it's a Key-Value store.

Very simple data models consisting of:

- Bucket Name
- Key Name
- Data
- Indexes

```
$ curl -XPUT "http://localhost:8098/riak/bucket/key"\
  --header "Content-Type: application/json" \
  --data '{"living_in":"the future!"}'

$ curl -XGET "http://localhost:8098/riak/bucket/key"
  {"living_in":"the future!"}
```

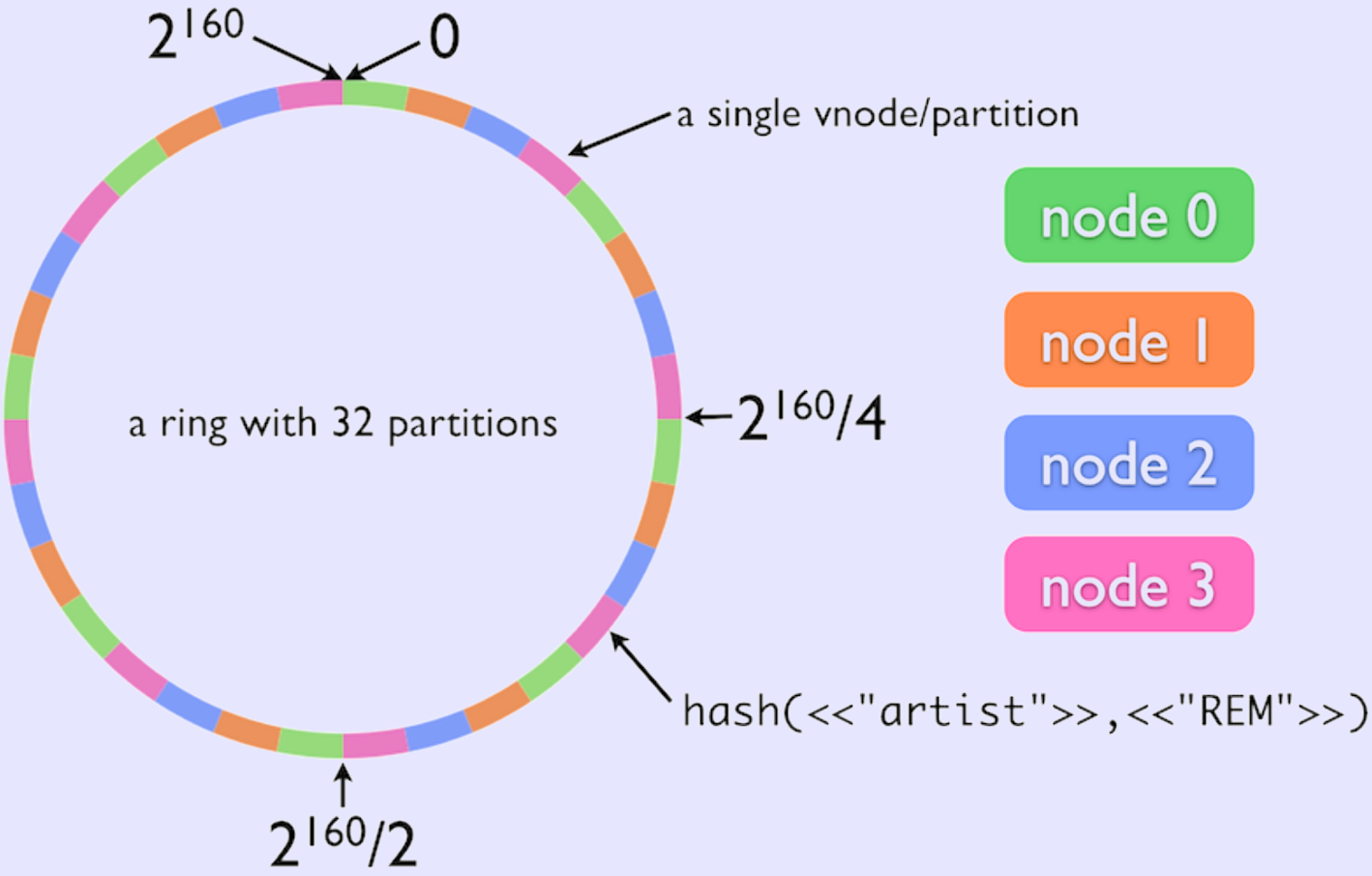
# Binary Data

```
$ curl -XPUT "http://localhost:8098/riak/images/horse.jpg"\
  --header "Content-Type: image/jpeg" \
  --data-binary @/home/user/horse_pic.jpg

$ curl -XGET "http://localhost:8098/riak/images/horse.jpg" >
/home/user/new_horse.jpg

$ md5sum /home/user/horse_pic.jpg
3f9bdc9366aec1f98839f47717ada5bf  horse_pic.jpg

$ md5sum /home/user/new_horse.jpg
3f9bdc9366aec1f98839f47717ada5bf  new_horse.jpg
```

# The Ring


Source: http://lotr.wikia.com/wiki/File:One_Ring_To_Rule_Them_All

# Distributed / Clustered




Source: http://docs.basho.com/shared/1.4.2/images/riak-ring.png?1383243838

# Horizontal Scaling

Simply add additional nodes to the ring:

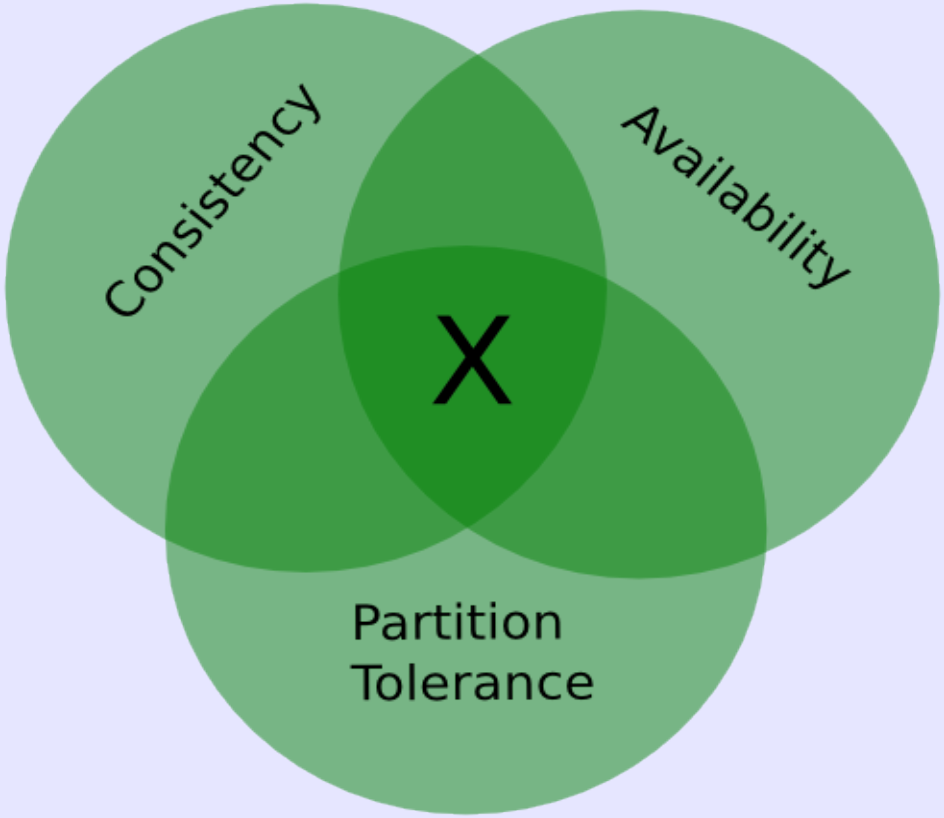
```
$ riak-admin cluster join riak@my_other_node.net
```

This causes no downtime and does not require you to reshard your data.

# CAP Theorem

“If you have a system that can get a network partition you have a choice: do you want to be consistent or do you want to be available?”

- Martin Fowler




Where does Riak fit?

# CAP Tuning

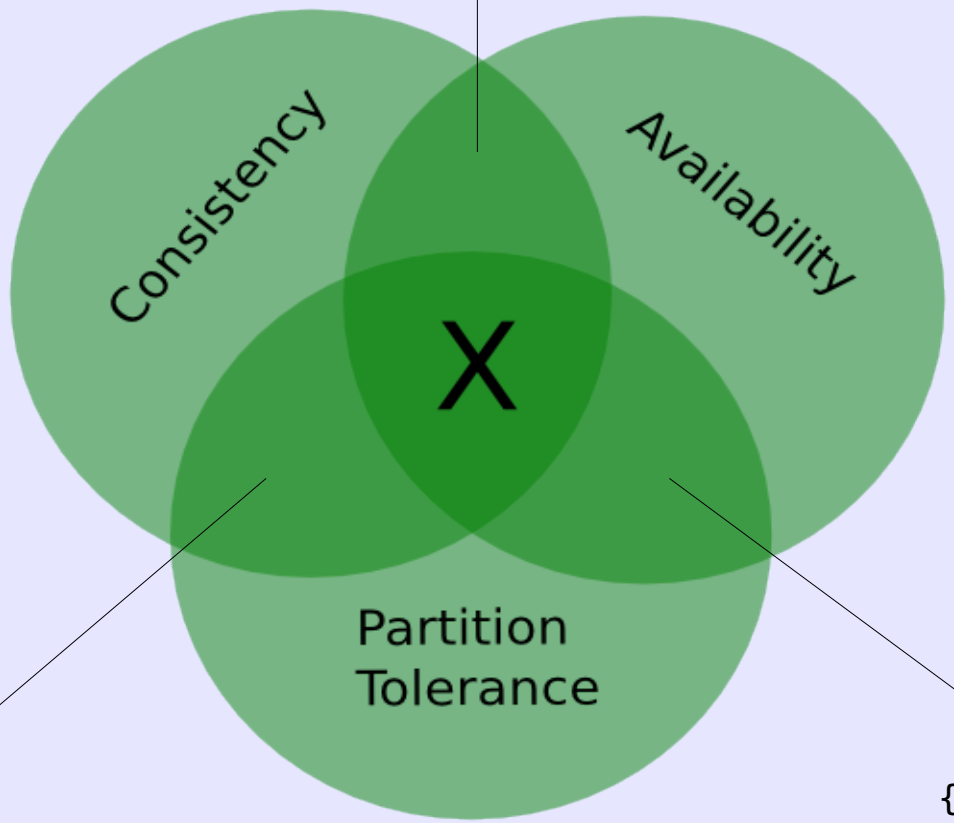
Request based CAP tuning via three primary properties:

- **N:** Copies of each key stored
- **R:** Number of nodes which must agree on a value for a read
- **W:** Number of nodes that must confirm receipt for a write

There are also two advanced options: **DW** and **RW.**

# Customize N,R,W to meet your CAP needs

```
{
  “n_val”: <all>,
  “r_val”: 1,
  “w_val”: <all>
}
```



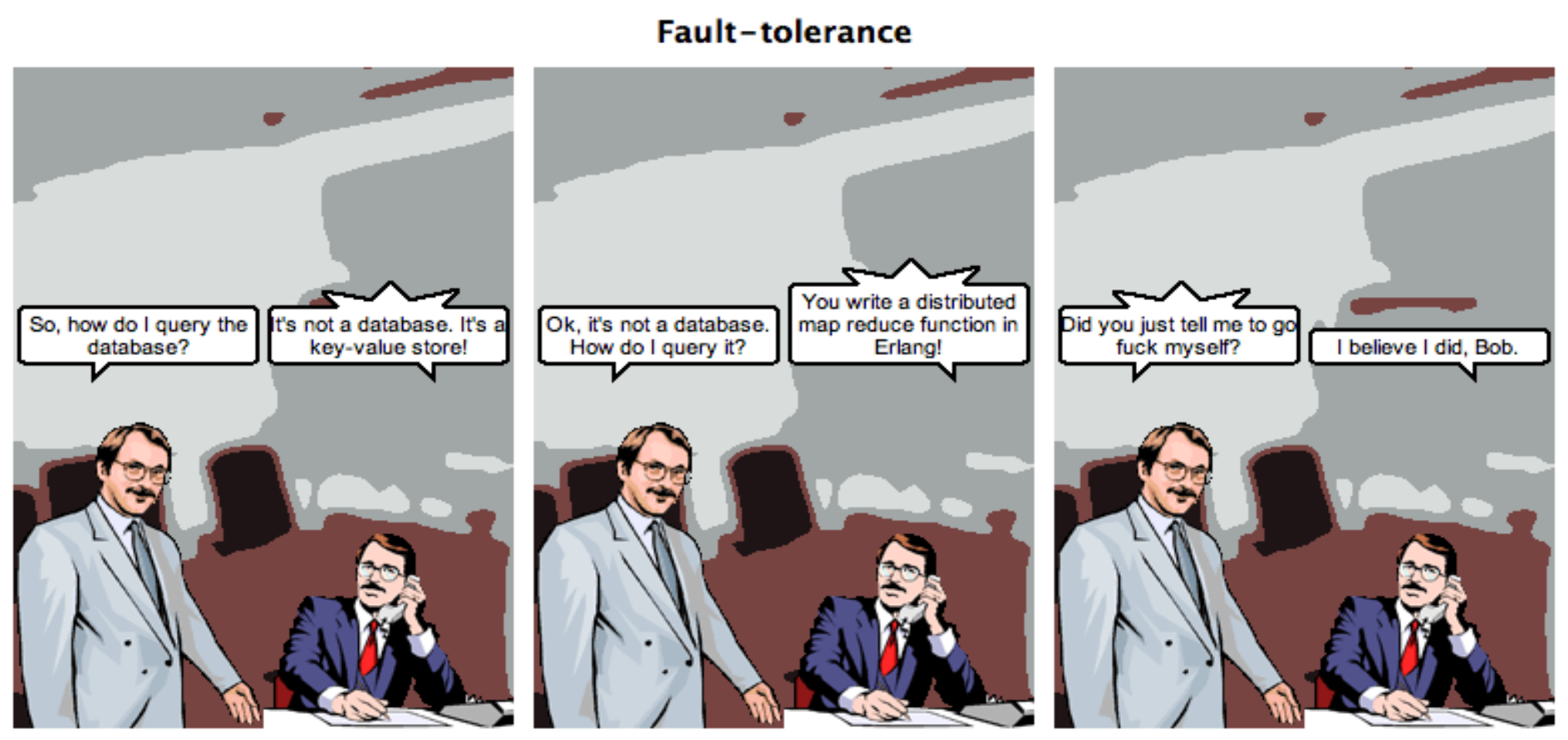

```
{
  “n_val”: <high value>,
  “r_val”: <n_val quorum+>,
  “w_val”: <n_val quorum+>
}
```

```
{
  “n_val”: <high value>,
  “r_val”: <low value>,
  “w_val”: <low value>
}
```

# Sample 1 – riak-client gem

```
require 'rubygems'
require 'riak'

# Create a client for localhost
@client = Riak::Client.new

# Specify a bucket
@bucket = @client.bucket('sample-bucket')

# Build an object
@object = @bucket.get_or_new('sample-key')
@object.content_type = 'application/json'
@object.data = '{"sample":"data"}'
@object.store

# Access the object
@bucket.get('sample-key')
#=> #<Riak::RObject {sample-bucket,sample-key} [#<Riak::RContent \
[application/json]:"{\"sample\":\"data\"}">]>
```

# Sample 2 – Map/Reduce




source: http://sistrix.net/19

# Map/Reduce Data Seed

```
$ curl -XPUT "http://localhost:8098/riak/users/btables@gmail.com"\
  --header "Content-Type: application/json" \
  --data '{
           "first_name":"Bobby",
           "last_name":"Tables"
          }'
```

For this example, assume we have many records like this in the 'users' bucket.

# Map/Reduce Request

```
$ curl -XPOST "http://localhost:8098/mapred"\
  --header "Content-Type: application/json" \
  --data '{
            "inputs":"users",
              "query":[{
              "map":{
                "language":"javascript",
                "source":"function(record) {
                   var record_data = JSON.parse(record.values[0].data);
                   return [
                     [
                       record.key,
                       record_data.first_name,
                       record_data.last_name
                     ]
                   ];
                 }"}}]
            }'

# Returns:
[["gabe@gabekoss.com","Gabe","Koss"],
["sam@stelfox.net","Sam","Stelfox"],
["btables@gmail.com","Bobby","Tables"]]
```

# What *shouldn't* I use Riak for?

- Key listing / Bucket enumeration (MyRiakObject.all() == :bad)
- Mass deletion from certain backends
- Large map-reduce in real-time operations

# What *should* I use Riak for?

- Clustered operations
- Straight key-value gets & puts
- Sibling resolution / link walking

# Resources

- Basho Technologies - https://basho.com/
- Sean Cribbs - @seancribbs
- Little Riak Book - http://littleriakbook.com/
- Learn You Some Erlang for Great Good – http://learnyousomeerlang.com/

```
$ curl -v -XGET "http://localhost:8098/riak/presentations/riak"

* About to connect() to localhost port 8098 (#0)
*   Trying 127.0.0.1... connected
> GET /riak/presentations/riak HTTP/1.1
> User-Agent: curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0
OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtmp/2.3
> Host: localhost:8098
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Riak-Vclock: a85hYGBgzGDKBVIcypz/fgauOPIqgymRPY+VQe7kv9N8WQA=
< Vary: Accept-Encoding
< Server: MochiWeb/1.1 WebMachine/1.9.2 (someone had painted it blue)
< Link: </riak/presentations>; rel="up"
< Last-Modified: Mon, 18 Nov 2013 17:23:42 GMT
< ETag: "7GBSRHNtXWBpYtpuv56JHT"
< Date: Mon, 18 Nov 2013 17:35:46 GMT
< Content-Type: application/json
< Content-Length: 131
<
  {
    "subject": "Riak",
    "presenters": [
      ["Sam Stelfox", "@SamStelfox", "http://stelfox.net" ],
      ["Gabe Koss",   "@granolocks", "http://gabekoss.com"]
    ]
  }
```