Hello User,
Welcome to Database

Joe DeVivo works for Basho,
tweets at @joedevivo, and
put some relevant files at github.com/joedevivo/ricon

If people can’t set up your product,
they secretly hate you

I found an amazing database back in 2010.

It wasn't too hard to configure back then

turns out, that was the only part that didn't scale

```
%% -*- tab-width: 4;erlang-indent-level: 4;indent-tabs-mode: nil -*-
%% ex: ts=4 sw=4 et
[
 %% Riak Core config
 {riak_core, [
    %% Default location of ringstate
    {ring_state_dir, "data/ring"},

    %% riak_web_ip is the IP address that Riak's HTTP interface will
    %%  bind to.  If this is undefined, the HTTP interface will not run.
    {web_ip, "127.0.0.1"},

    %% riak_web_port is the TCP port that Riak's HTTP interface will
    %% bind to.
     {web_port, 8098}
    ]},

%% Riak KV config
{riak_kv, [
    %% Storage_backend specifies the Erlang module defining the storage
    %% mechanism that will be used on this node.
    {storage_backend, riak_kv_dets_backend},

    %% Different storage backends can use other configuration variables.
    %%  For instance, riak_dets_backend_root determines the directory
    %%  under which dets files will be placed.
    {riak_kv_dets_backend_root, "data/dets"},
```

```
%% -*- tab-width: 4;erlang-indent-level: 4;indent-tabs-mode: nil -*-
%% ex: ts=4 sw=4 et
[
 %% Riak Core config
 {riak_core, [
             %% Default location of ringstate
             {ring_state_dir, "data/ring"},

             %% riak_web_ip is the IP address that Riak's HTTP interface will
             %%  bind to.  If this is undefined, the HTTP interface will not run.
             {web_ip, "127.0.0.1"},

             %% riak_web_port is the TCP port that Riak's HTTP interface will
             %% bind to.
             {web_port, 8098}
            ]},

 %% Riak KV config
 {riak_kv, [
           %% Storage_backend specifies the Erlang module defining the storage
           %% mechanism that will be used on this node.
           {storage_backend, riak_kv_dets_backend},

           %% Different storage backends can use other configuration variables.
           %%  For instance, riak_dets_backend_root determines the directory
           %%  under which dets files will be placed.
           {riak_kv_dets_backend_root, "data/dets"},

           %% riak_handoff_port is the TCP port that Riak uses for
           %% intra-cluster data handoff.
           {handoff_port, 8099},

           %% pb_ip is the IP address that Riak's Protocol Buffers interface
           %% will bid to.  If this is undefined, the interface will not run.
           {pb_ip,   "0.0.0.0"},

           %% pb_port is the TCP port that Riak's Protocol Buffers interface
           %% will bind to
           {pb_port, 8087},

           %% raw_name is the first part of all URLS used by Riak's raw HTTP
           %% interface.  See riak_web.erl and raw_http_resource.erl for
           %% details.
           %{raw_name, "riak"},

           %% mapred_name is URL used to submit map/reduce requests to Riak.
           {mapred_name, "mapred"},

           %% js_vm_count is the number of Javascript VMs to start per Riak
           %% node.  8 is a good default for smaller installations. A larger
           %% number like 12 or 16 is appropriate for installations handling
           %% lots of map/reduce processing.
           {js_vm_count, 8},

           %% js_source_dir should point to a directory containing Javascript
           %% source files which will be loaded by Riak when it initializes
           %% Javascript VMs.
           %{js_source_dir, "/tmp/js_source"}

           %% riak_stat enables the use of the "riak-admin status" command to
           %% retrieve information the Riak node for performance and debugging needs
           {riak_kv_stat, true}
          ]},

 %% SASL config
 {sasl, [
         {sasl_error_logger, {file, "log/sasl-error.log"}},
         {errlog_type, error},
         {error_logger_mf_dir, "log/sasl"},      % Log directory
         {error_logger_mf_maxbytes, 10485760},   % 10 MB max file size
         {error_logger_mf_maxfiles, 5}           % 5 files max
         ]}
].
```

```
## Name of the riak node
-name dev1@127.0.0.1

## Cookie for distributed erlang
-setcookie riak

## Heartbeat management; auto-restarts VM if it dies or becomes unresponsive
## (Disabled by default..use with caution!)
##-heart

## Enable kernel poll and a few async threads
+K true
+A 5

## Increase number of concurrent ports/sockets
-env ERL_MAX_PORTS 4096

## Tweak GC to run more often
-env ERL_FULLSWEEP_AFTER 10
```

Two configuration files with different syntaxes!

That's sure to win us database of the year!

#sarcasm

Guess how many times I broke this?

A lot

Forget a comma?

Your configuration broke

Didn't close that bracket?

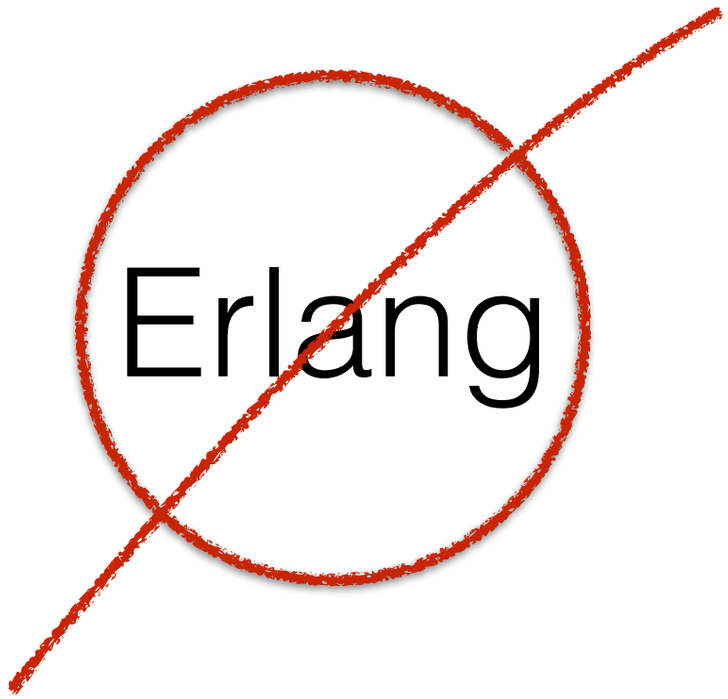
Too bad

Not an Erlang developer?

Well, you should be

Erlang

I've got hard distributed
datab@ss problems to solve


Erlang

LOL, Syntax

Erlang developers write Erlang friendly configuration settings

- `{on, []}`
- `{off, []}`
- `{on, [debug]}`

one setting, lots of lines

```
%% cool syntax, bro
[{riak_kv, [
        {anti_entropy,
           {on, [debug]}}
        ]}
].
```

actual number!

app.config

hyperbole!

warning: contains about a billion

[]’s, {}’s, commas, and periods to misplace

NOTHING!

Everything you need to know about a single configuration setting should be on a single line

That line can be anywhere in the file

If you set something more than once,
the last one wins

Shoutout to
Last Write Wins!

Key/Value is a pretty good idea

What did I want to give the user?

```
## enable active anti-entropy subsystem
## possible values: on, off, debug
anti_entropy = on
```

cuttlefish

```
## Restrict how fast AAE can build hash trees. Building the tree
## for a given partition requires a full scan over that partition's
## data. Once built, trees stay built until they are expired.
## Config is of the form:
## {num-builds, per-timespan}
## Default is 1 build per hour.
anti_entropy.build_limit.number = 1

anti_entropy.build_limit.per_timespan = 1h

## Determine how often hash trees are expired after being built.
## Periodically expiring a hash tree ensures the on-disk hash tree
## data stays consistent with the actual k/v backend data. It also
## helps Riak identify silent disk failures and bit rot. However,
## expiration is not needed for normal AAE operation and should be
## infrequent for performance reasons. The time is specified in
## milliseconds. The default is 1 week.
anti_entropy.expire = 1w

## Limit how many AAE exchanges/builds can happen concurrently.
anti_entropy.concurrency = 2

## The tick determines how often the AAE manager looks for work
## to do (building/expiring trees, triggering exchanges, etc).
## The default is every 15 seconds. Lowering this value will
## speedup the rate that all replicas are synced across the cluster.
## Increasing the value is not recommended.
anti_entropy.tick = 15s
```

```
➜  rel git:(develop) ✗ cat riak.conf | grep anti_entropy
anti_entropy = on
anti_entropy.build_limit.number = 1
anti_entropy.build_limit.per_timespan = 1h
anti_entropy.expire = 1w
anti_entropy.concurrency = 2
anti_entropy.tick = 15s
anti_entropy.data_dir = ./data/anti_entropy
anti_entropy.write_buffer_size = 4MB
anti_entropy.max_open_files = 20
```

```
## Default ring creation size.  Make sure it is a power of 2,
## e.g. 16, 32, 64, 128, 256, 512 etc
ring_size = 42
```

```
➜  rel git:(develop) ✗ ./riak/bin/riak console
10:11:27.284 [info] Application lager started on node nonode@nohost
10:11:27.284 [info] Checking /Users/joe/dev/basho/riak/rel/riak/bin/../
etc/app.config exists... false
10:11:27.284 [info] Checking /Users/joe/dev/basho/riak/rel/riak/bin/../
etc/vm.args exists... false
10:11:27.284 [info] No app.config or vm.args detected in /Users/joe/dev/
basho/riak/rel/riak/bin/../etc, activating cuttlefish
10:11:27.717 [info] Adding Defaults
10:11:27.723 [info] Applying Datatypes
10:11:27.751 [info] Validation
10:11:27.751 [error] ring_size invalid, not a power of 2 greater than 1
10:11:27.751 [error] Some validator failed, aborting
```

vm.args

app.config

1.4.2

riak.conf

2.0

Thanks