Building Fault Tolerant Teams At Basho*

Reid Draper @reiddraper

*An unexpected journey

thanks Garrett

Garrett helped write my talk description…

Hip hip
Erlang!

Black space created by 4:3 aspect ratio
(better)

So blame him if you don't get your money's worth

We're *capable* of building fault-tolerant applications with Erlang

But Erlang is not *sufficient*

A *human* still has to write the software

What tooling can we use to augment our brains?

What tooling can we use to augment our teams?

Especially when we have a mission-critical database written by thirty developers

bors buildbot
code-review
dialyzer edoc
eunit quickcheck
riak-test xref

Basho and Riak

Riak is a distributed, fault-tolerant database maximum availability

no data-loss

How did we get here?

How did this bug make it through our process?

And can we improve our process to prevent similar bugs in the future?

Is it a logic issue?

A type-mismatch?

A performance issue?

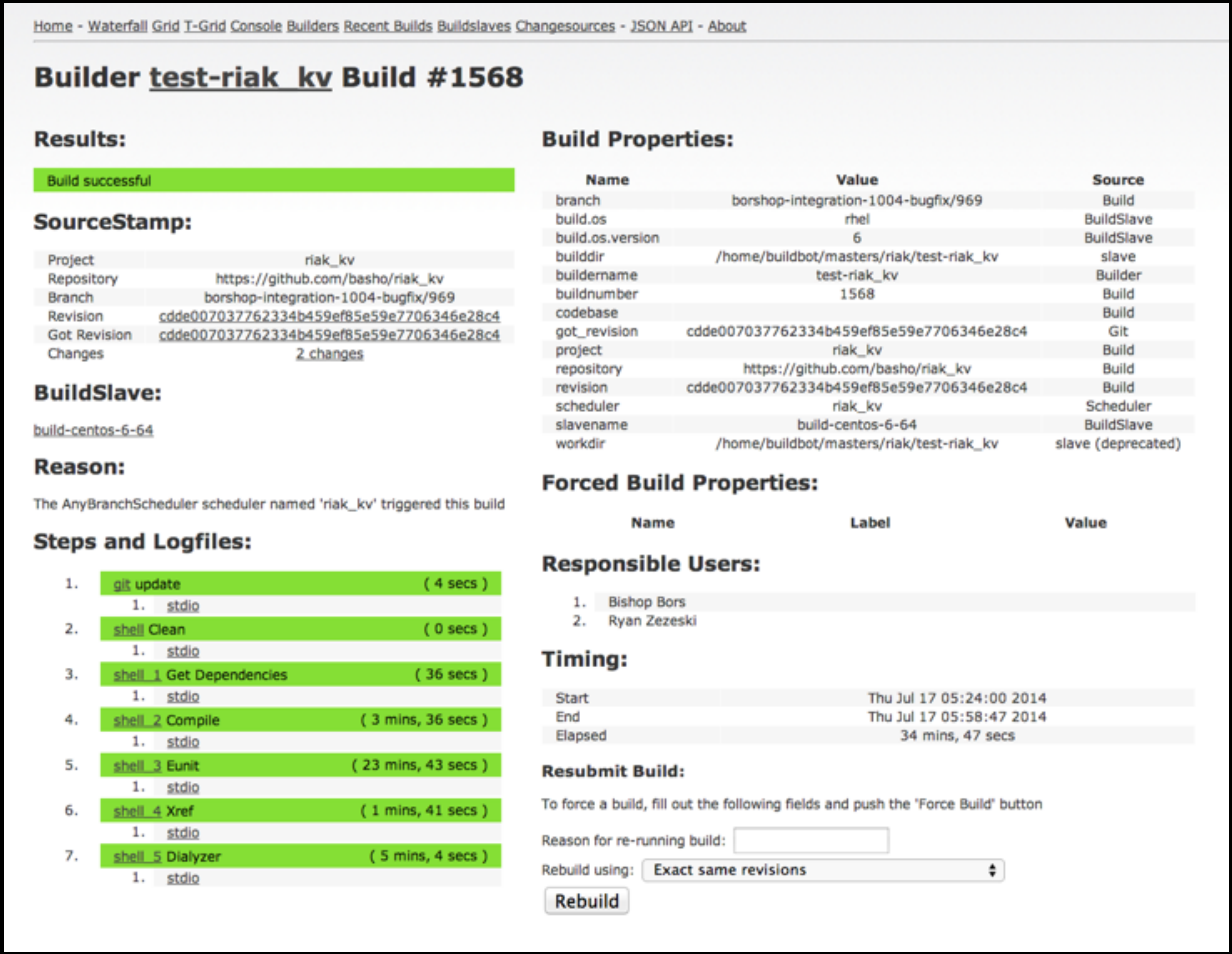
A typo?

bors buildbot
code-review
dialyzer edoc
eunit quickcheck
riak-test xref

bors buildbot
code-review
dialyzer edoc
eunit quickcheck
riak-test xref

buildbot

CI framework


Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About

Builder test-riak_kv Build #1568

Results:

Build successful

SourceStamp:

Project	riak_kv
Repository	https://github.com/basho/riak_kv
Branch	borshop-integration-1004-bugfix/969
Revision	cdde007037762334b459ef85e59e7706346e28c4
Got Revision	cdde007037762334b459ef85e59e7706346e28c4
Changes	2 changes

BuildSlave:

build-centos-6-64

Reason:

The AnyBranchScheduler scheduler named 'riak_kv' triggered this build

Steps and Logfiles:

1. git update (4 secs)
   1. stdio
2. shell Clean (0 secs)
   1. stdio
3. shell_1 Get Dependencies (36 secs)
   1. stdio
4. shell_2 Compile (3 mins, 36 secs)
   1. stdio
5. shell_3 Eunit (23 mins, 43 secs)
   1. stdio
6. shell_4 Xref (1 mins, 41 secs)
   1. stdio
7. shell_5 Dialyzer (5 mins, 4 secs)
   1. stdio

Build Properties:

Name	Value	Source
branch	borshop-integration-1004-bugfix/969	Build
build.os	rhel	BuildSlave
build.os.version	6	BuildSlave
builddir	/home/buildbot/masters/riak/test-riak_kv	slave
buildername	test-riak_kv	Builder
buildnumber	1568	Build
codebase		Build
got_revision	cdde007037762334b459ef85e59e7706346e28c4	Git
project	riak_kv	Build
repository	https://github.com/basho/riak_kv	Build
revision	cdde007037762334b459ef85e59e7706346e28c4	Build
scheduler	riak_kv	Scheduler
slavename	build-centos-6-64	BuildSlave
workdir	/home/buildbot/masters/riak/test-riak_kv	slave (deprecated)

Forced Build Properties:

Name	Label	Value

Responsible Users:

1. Bishop Bors
2. Ryan Zezeski

Timing:

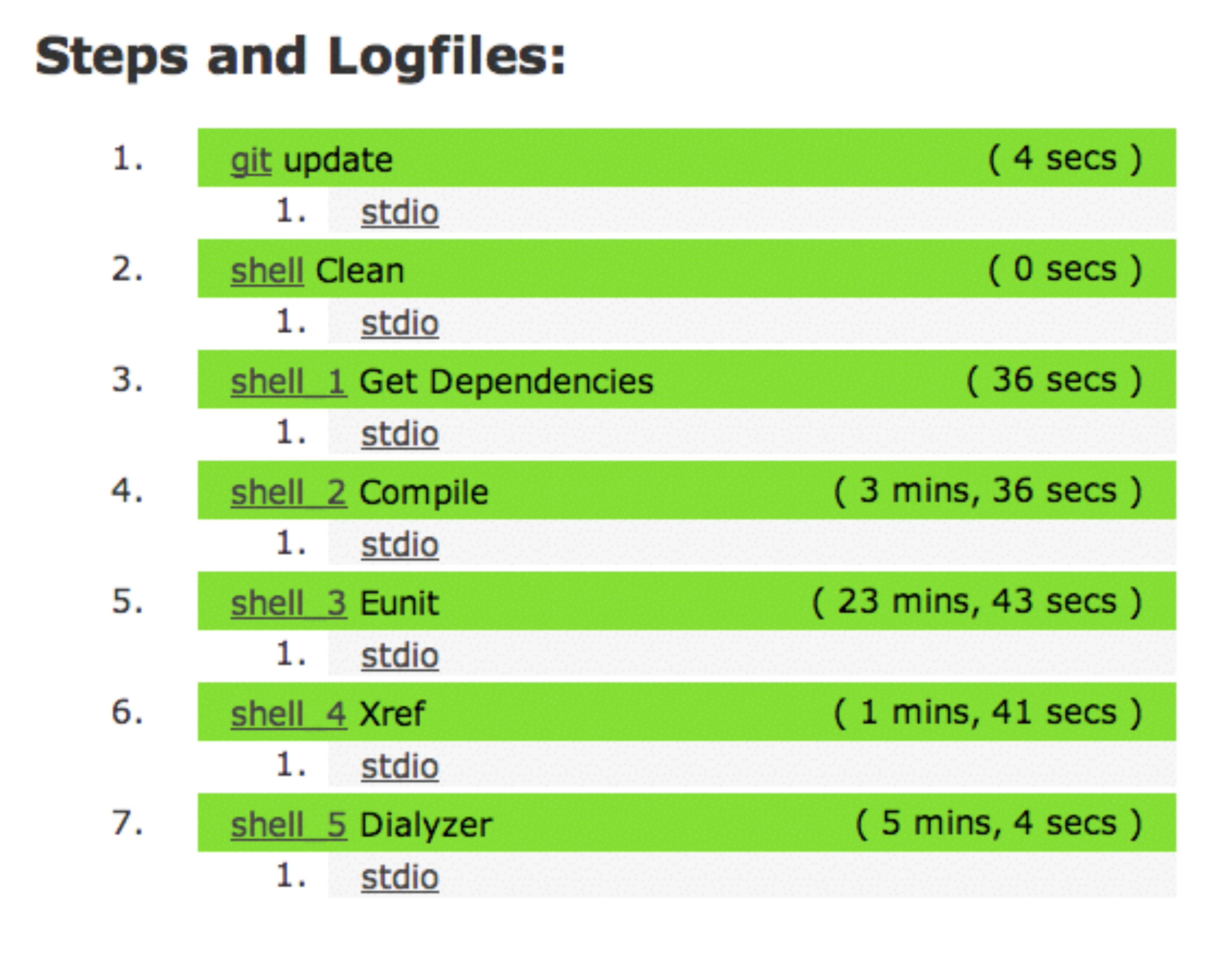
Start	Thu Jul 17 05:24:00 2014
End	Thu Jul 17 05:58:47 2014
Elapsed	34 mins, 47 secs

Resubmit Build:

To force a build, fill out the following fields and push the 'Force Build' button

Reason for re-running build:

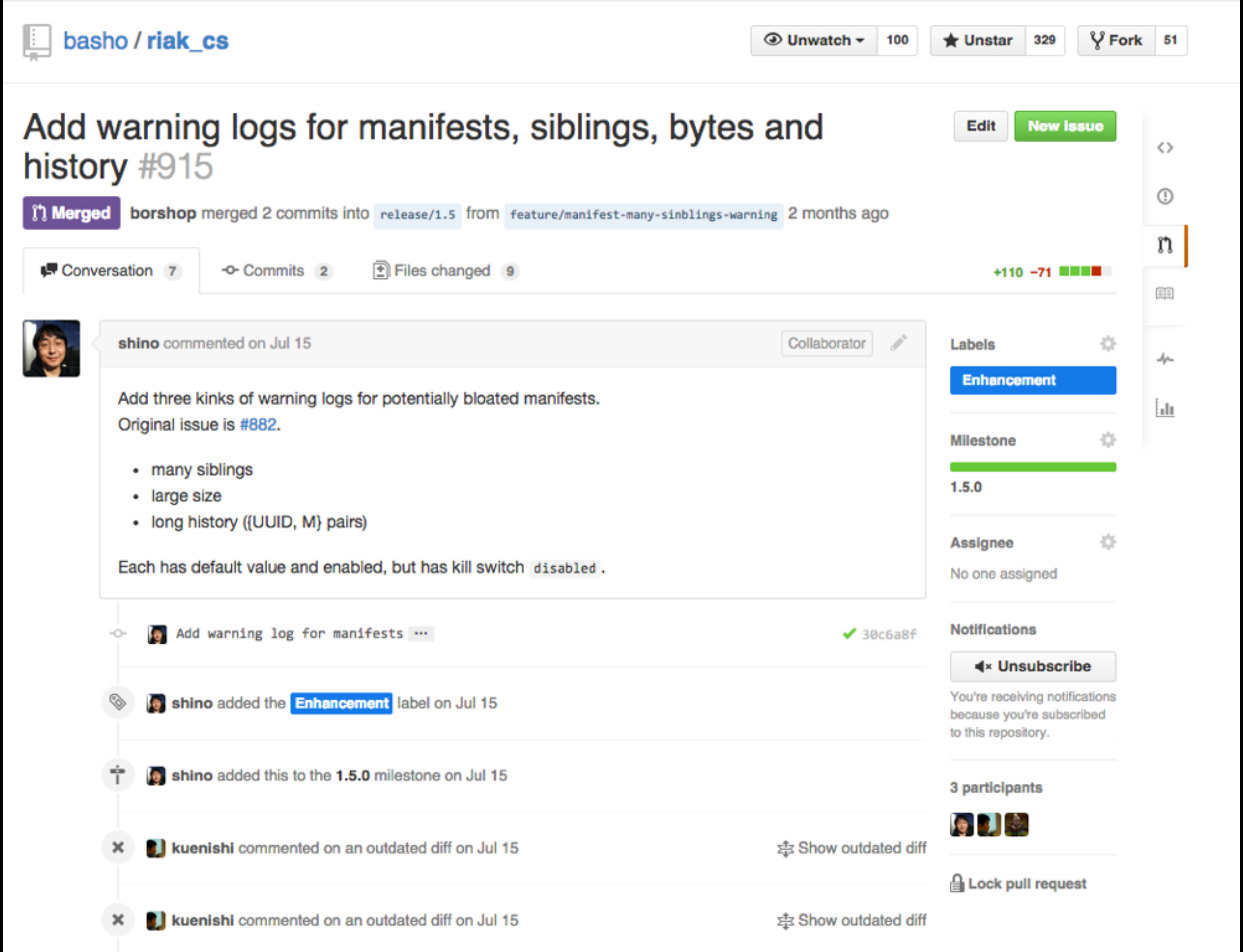
Rebuild using: Exact same revisions



Steps and Logfiles:

1. git update (4 secs)
   1. stdio
2. shell Clean (0 secs)
   1. stdio
3. shell_1 Get Dependencies (36 secs)
   1. stdio
4. shell_2 Compile (3 mins, 36 secs)
   1. stdio
5. shell_3 Eunit (23 mins, 43 secs)
   1. stdio
6. shell_4 Xref (1 mins, 41 secs)
   1. stdio
7. shell_5 Dialyzer (5 mins, 4 secs)
   1. stdio

code-review



Add warning logs for manifests, siblings, bytes and history #915

Merged borshop merged 2 commits into `release/1.5` from `feature/manifest-many-sinblings-warning` 2 months ago

shino commented on Jul 15

Add three kinks of warning logs for potentially bloated manifests.
Original issue is #882.

- many siblings
- large size
- long history ({UUID, M} pairs)

Each has default value and enabled, but has kill switch `disabled`.

`Add warning log for manifests …` ✓ 30c6a8f

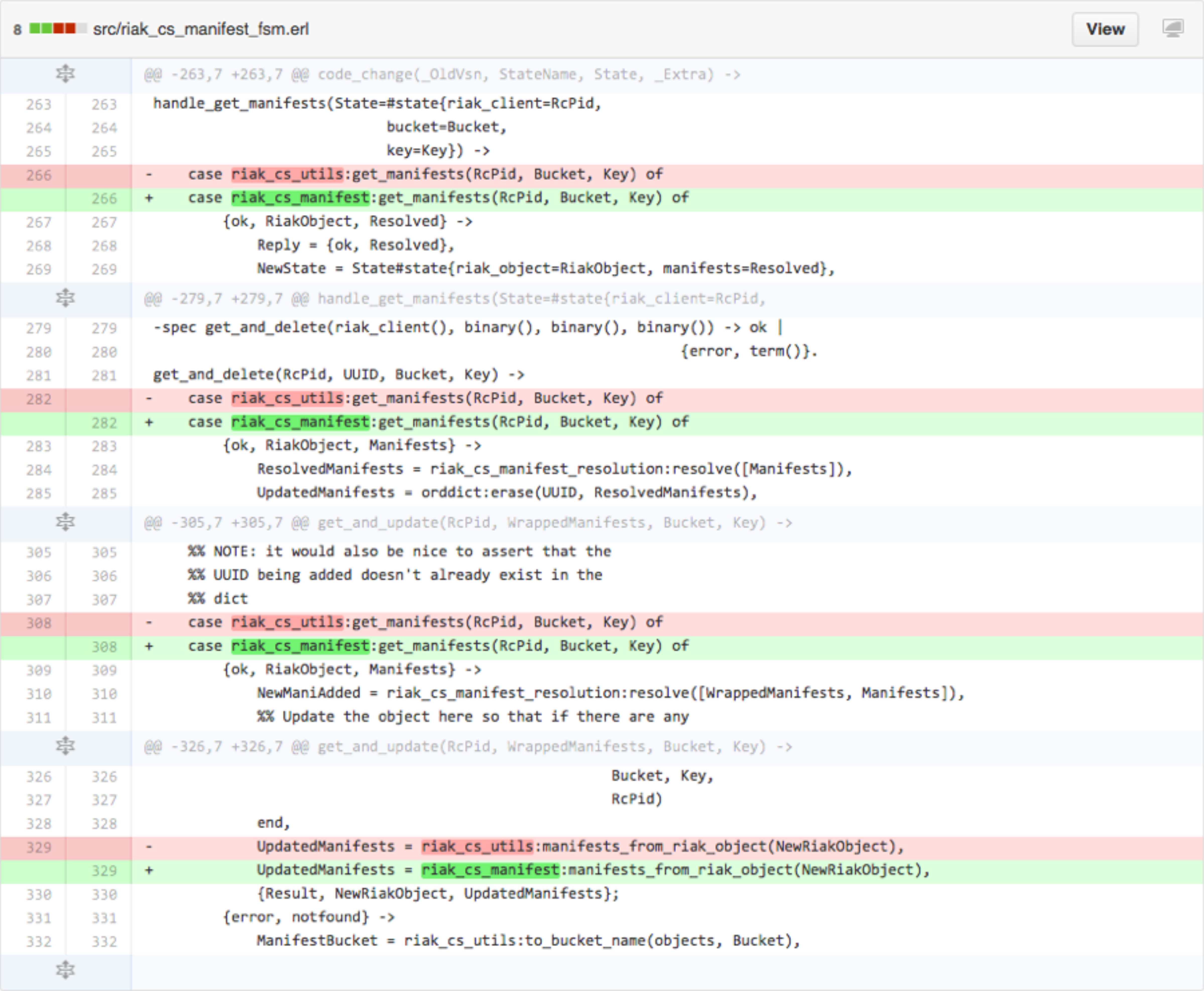
shino added the Enhancement label on Jul 15

shino added this to the **1.5.0** milestone on Jul 15

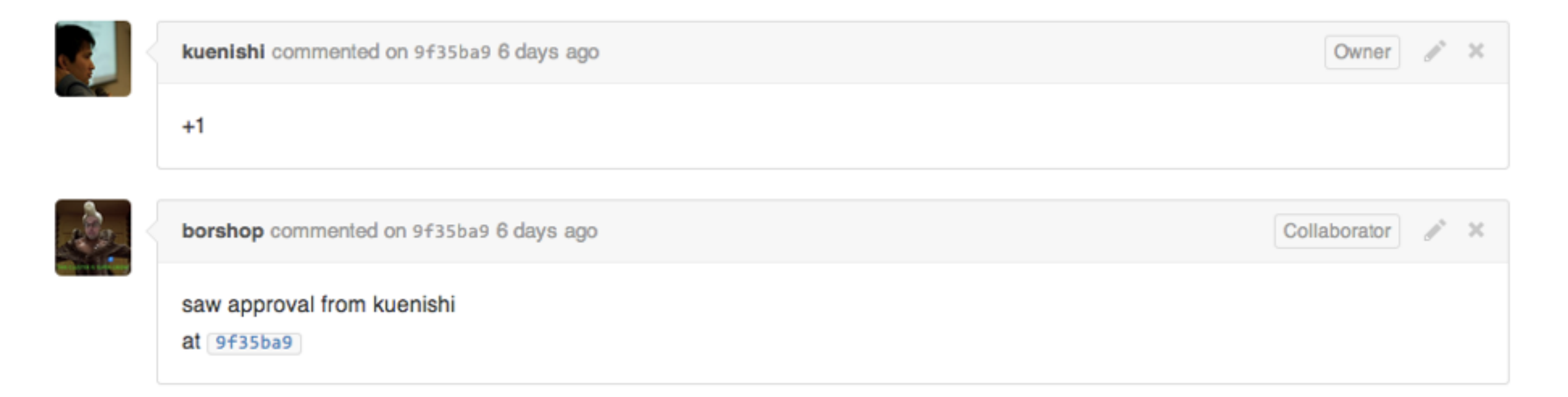
kuenishi commented on an outdated diff on Jul 15

kuenishi commented on an outdated diff on Jul 15

You're receiving notifications because you're subscribed to this repository.



```
@@ -263,7 +263,7 @@ code_change(_OldVsn, StateName, State, _Extra) ->
 handle_get_manifests(State=#state{riak_client=RcPid,
                            bucket=Bucket,
                            key=Key}) ->
-    case riak_cs_utils:get_manifests(RcPid, Bucket, Key) of
+    case riak_cs_manifest:get_manifests(RcPid, Bucket, Key) of
         {ok, RiakObject, Resolved} ->
             Reply = {ok, Resolved},
             NewState = State#state{riak_object=RiakObject, manifests=Resolved},
@@ -279,7 +279,7 @@ handle_get_manifests(State=#state{riak_client=RcPid,
 -spec get_and_delete(riak_client(), binary(), binary(), binary()) -> ok |
                                                              {error, term()}.
 get_and_delete(RcPid, UUID, Bucket, Key) ->
-    case riak_cs_utils:get_manifests(RcPid, Bucket, Key) of
+    case riak_cs_manifest:get_manifests(RcPid, Bucket, Key) of
         {ok, RiakObject, Manifests} ->
             ResolvedManifests = riak_cs_manifest_resolution:resolve([Manifests]),
             UpdatedManifests = orddict:erase(UUID, ResolvedManifests),
@@ -305,7 +305,7 @@ get_and_update(RcPid, WrappedManifests, Bucket, Key) ->
     %% NOTE: it would also be nice to assert that the
     %% UUID being added doesn't already exist in the
     %% dict
-    case riak_cs_utils:get_manifests(RcPid, Bucket, Key) of
+    case riak_cs_manifest:get_manifests(RcPid, Bucket, Key) of
         {ok, RiakObject, Manifests} ->
             NewManiAdded = riak_cs_manifest_resolution:resolve([WrappedManifests, Manifests]),
             %% Update the object here so that if there are any
@@ -326,7 +326,7 @@ get_and_update(RcPid, WrappedManifests, Bucket, Key) ->
                                                      Bucket, Key,
                                                      RcPid)
             end,
-            UpdatedManifests = riak_cs_utils:manifests_from_riak_object(NewRiakObject),
+            UpdatedManifests = riak_cs_manifest:manifests_from_riak_object(NewRiakObject),
             {Result, NewRiakObject, UpdatedManifests};
         {error, notfound} ->
             ManifestBucket = riak_cs_utils:to_bucket_name(objects, Bucket),
```



kuenishi commented on 9f35ba9 6 days ago

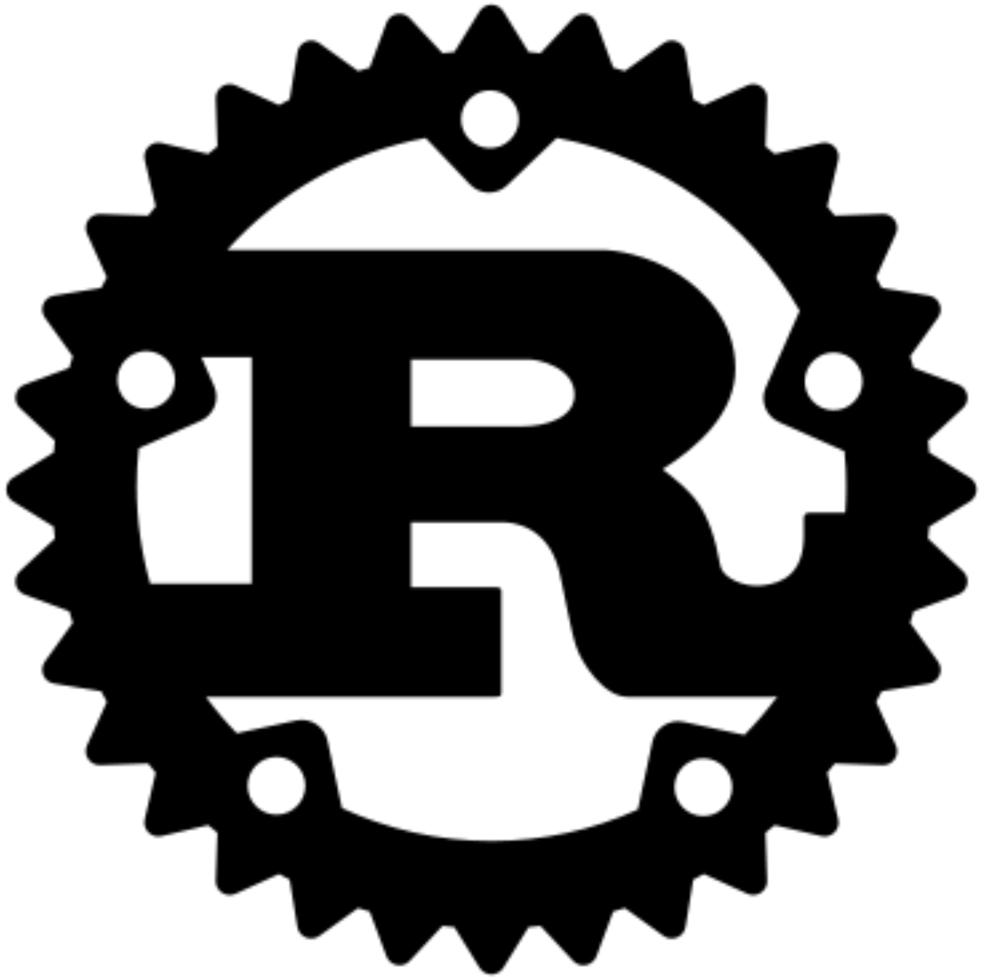
+1

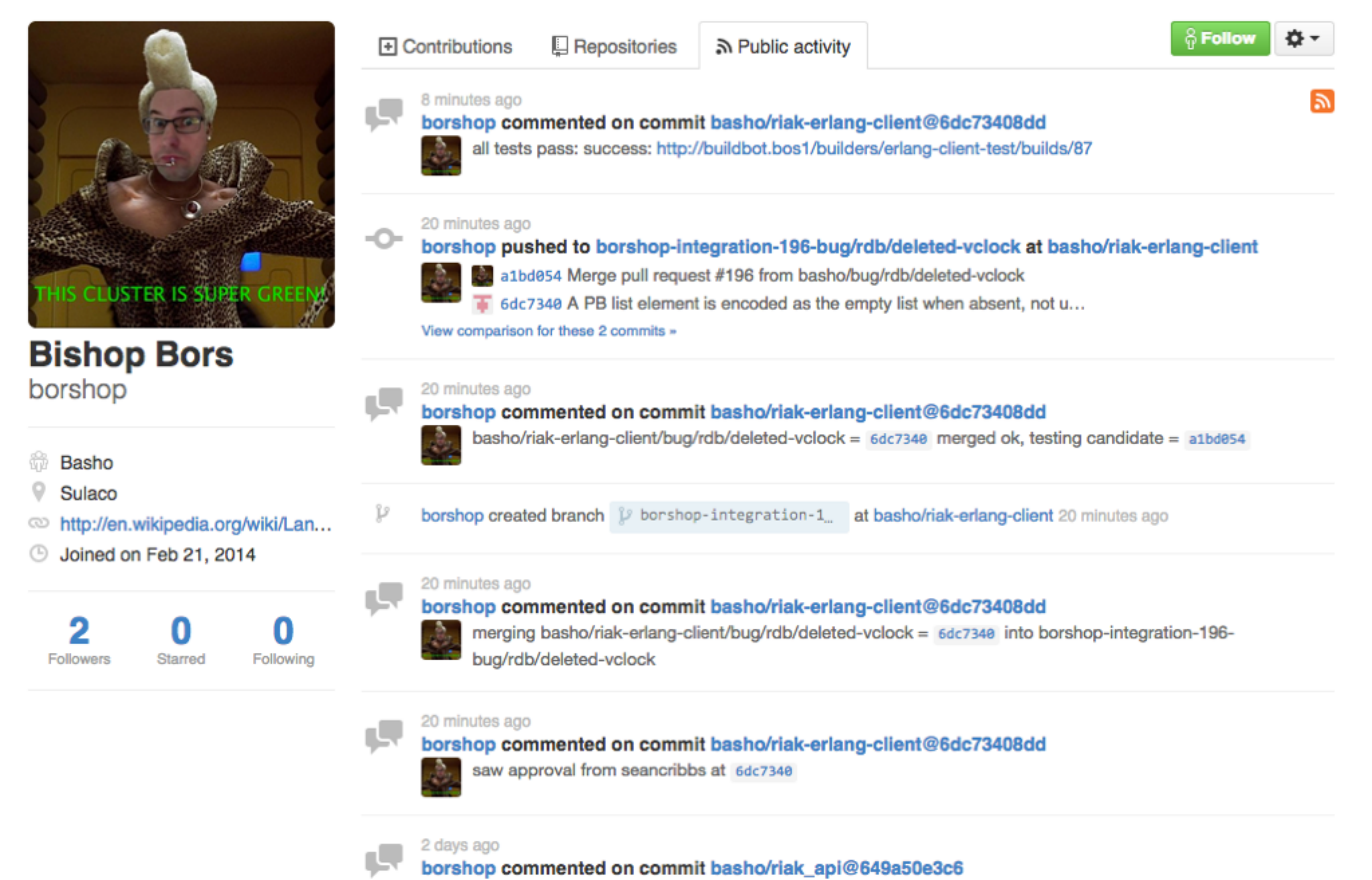
borshop commented on 9f35ba9 6 days ago

saw approval from kuenishi
at `9f35ba9`

bors

github.com/graydon/bors





Bishop Bors
borshop

Basho
Sulaco
http://en.wikipedia.org/wiki/Lan…
Joined on Feb 21, 2014

2	0	0
Followers	Starred	Following

8 minutes ago
borshop **commented on commit** basho/riak-erlang-client@6dc73408dd
all tests pass: success: http://buildbot.bos1/builders/erlang-client-test/builds/87

20 minutes ago
borshop **pushed to** borshop-integration-196-bug/rdb/deleted-vclock **at** basho/riak-erlang-client
`a1bd054` Merge pull request #196 from basho/bug/rdb/deleted-vclock
`6dc7340` A PB list element is encoded as the empty list when absent, not u…
View comparison for these 2 commits »

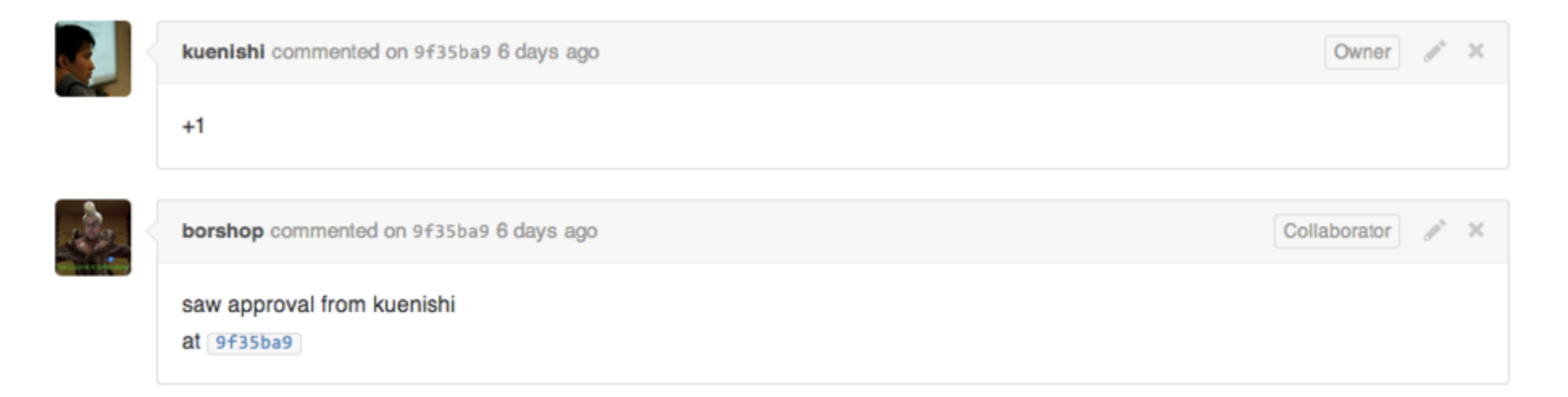
20 minutes ago
borshop **commented on commit** basho/riak-erlang-client@6dc73408dd
basho/riak-erlang-client/bug/rdb/deleted-vclock = `6dc7340` merged ok, testing candidate = `a1bd054`

borshop created branch `borshop-integration-1…` at basho/riak-erlang-client 20 minutes ago

20 minutes ago
borshop **commented on commit** basho/riak-erlang-client@6dc73408dd
merging basho/riak-erlang-client/bug/rdb/deleted-vclock = `6dc7340` into borshop-integration-196-bug/rdb/deleted-vclock

20 minutes ago
borshop **commented on commit** basho/riak-erlang-client@6dc73408dd
saw approval from seancribbs at `6dc7340`

2 days ago
borshop **commented on commit** basho/riak_api@649a50e3c6

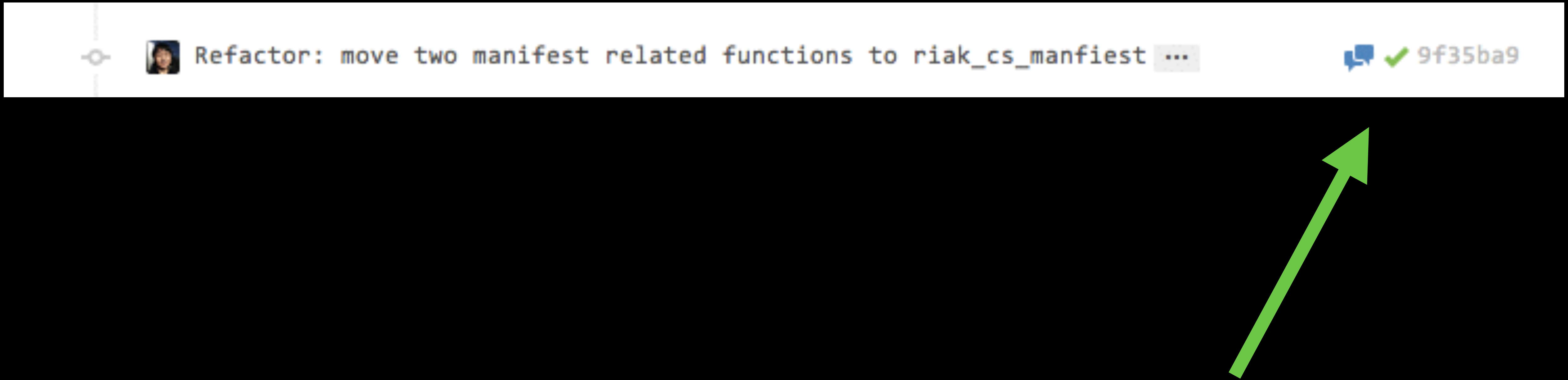


kuenishi commented on `9f35ba9` 6 days ago

+1

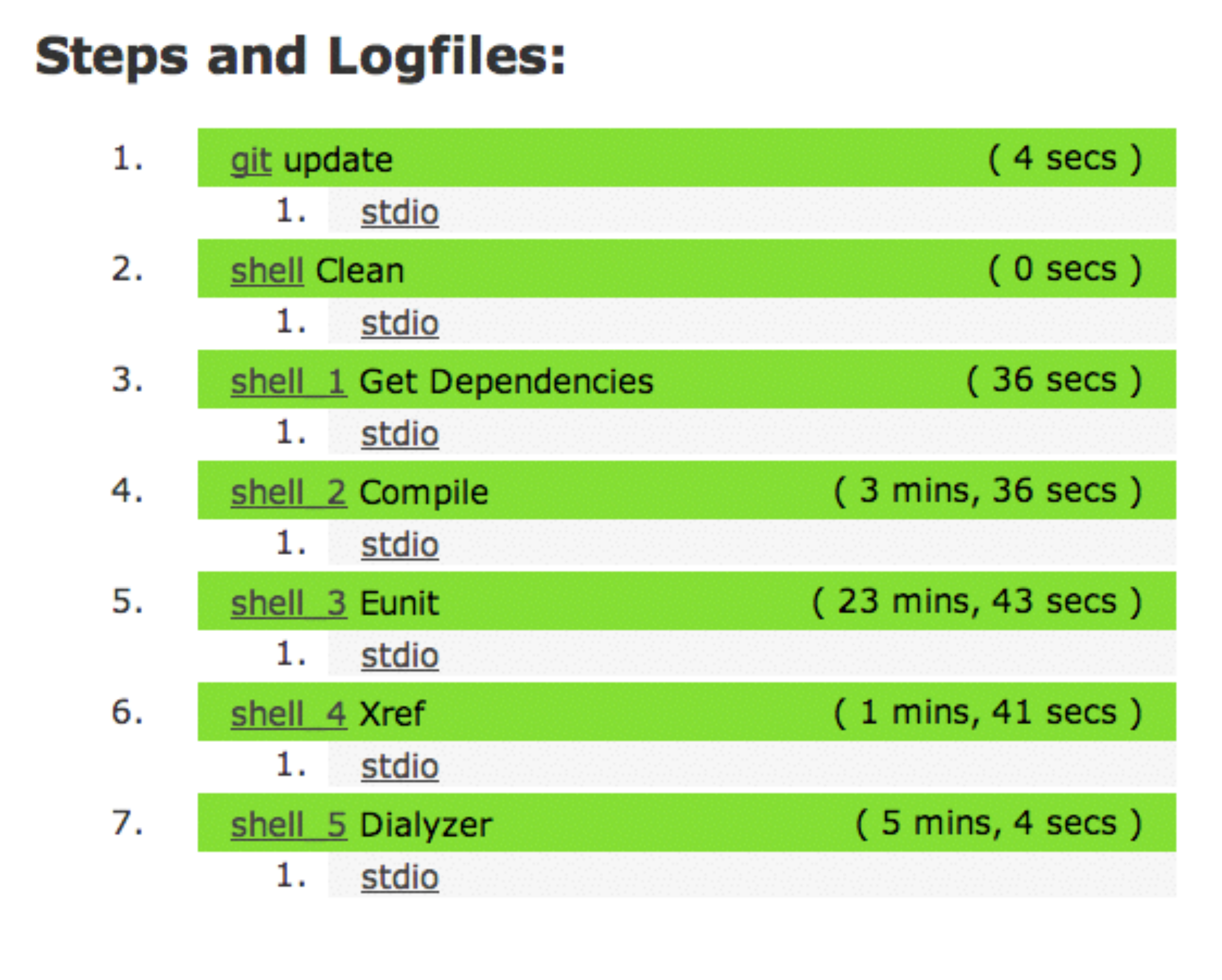
borshop commented on `9f35ba9` 6 days ago

saw approval from kuenishi
at `9f35ba9`



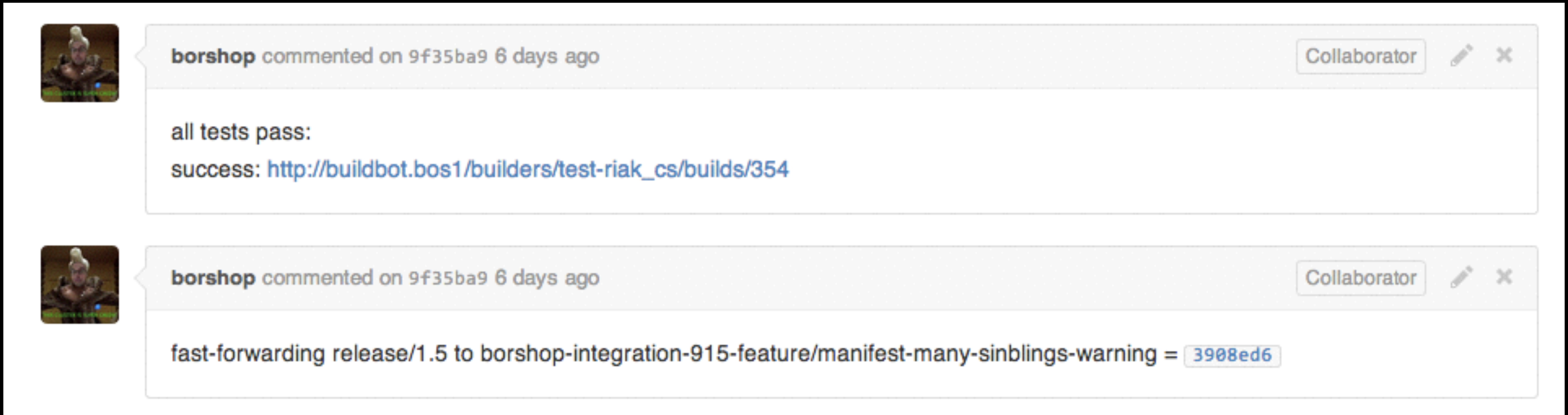
Refactor: move two manifest related functions to riak_cs_manfiest …

✓ 9f35ba9



Steps and Logfiles:

1. git update (4 secs)
    1. stdio
2. shell Clean (0 secs)
    1. stdio
3. shell_1 Get Dependencies (36 secs)
    1. stdio
4. shell_2 Compile (3 mins, 36 secs)
    1. stdio
5. shell_3 Eunit (23 mins, 43 secs)
    1. stdio
6. shell_4 Xref (1 mins, 41 secs)
    1. stdio
7. shell_5 Dialyzer (5 mins, 4 secs)
    1. stdio



borshop commented on `9f35ba9` 6 days ago

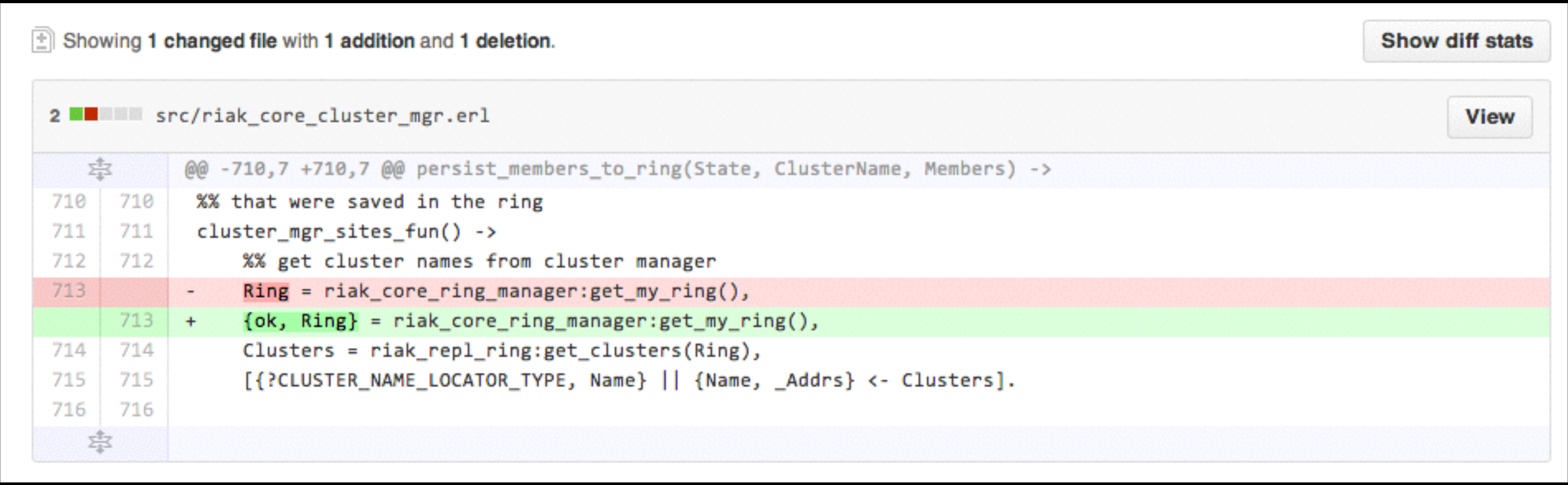
all tests pass:
success: http://buildbot.bos1/builders/test-riak_cs/builds/354

borshop commented on `9f35ba9` 6 days ago

fast-forwarding release/1.5 to borshop-integration-915-feature/manifest-many-sinblings-warning = `3908ed6`

Dialyzer

```
%% Return a list of locators, in our case,
%% we'll use cluster names
%% that were saved in the ring
cluster_mgr_sites_fun() ->
    %% get cluster names from cluster manager
    Ring = riak_core_ring_manager:get_my_ring(),
    Clusters = riak_repl_ring:get_clusters(Ring),
    [{?CLUSTER_NAME_LOCATOR_TYPE, Name} || {Name, _Addrs} <-
Clusters].
```



Showing **1 changed file** with **1 addition** and **1 deletion**.

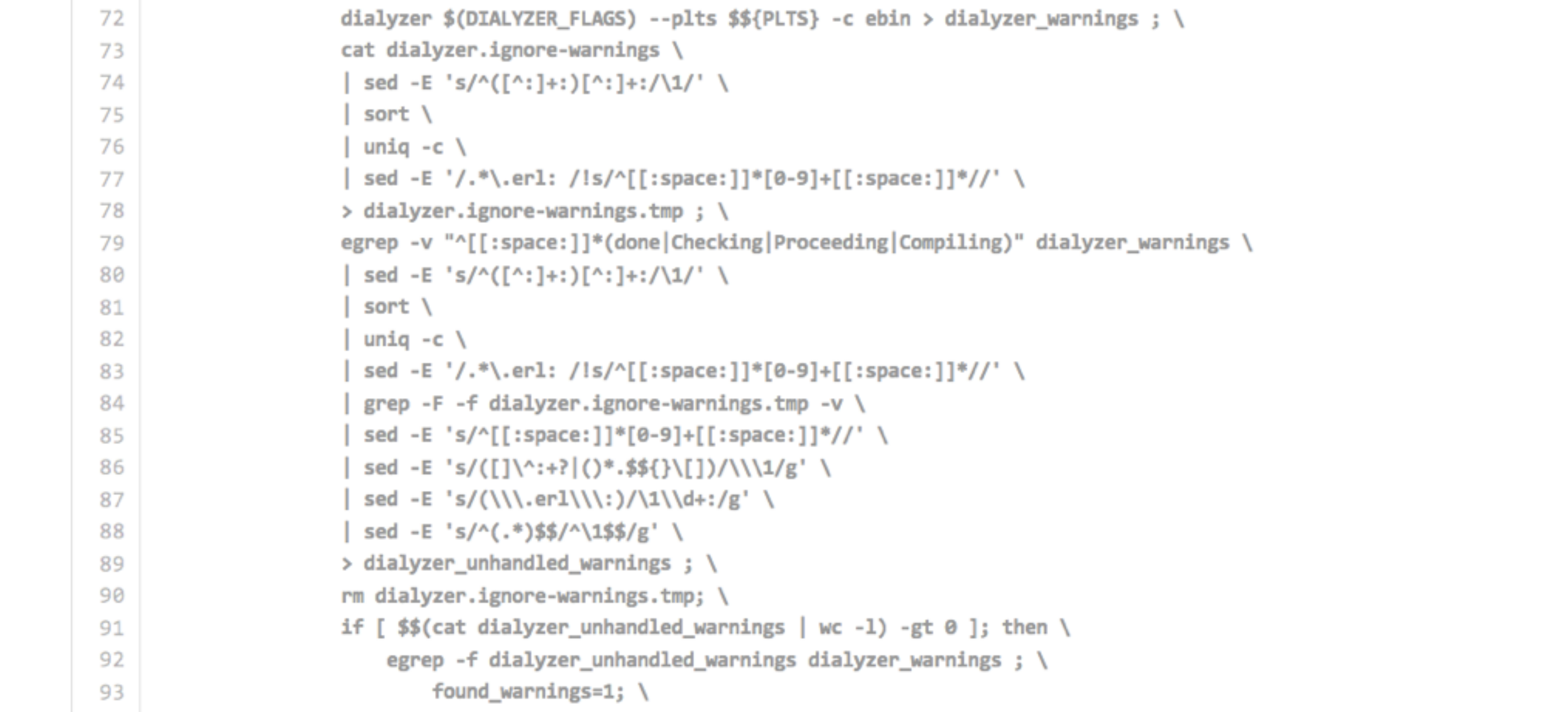
2 🟩🟥 src/riak_core_cluster_mgr.erl

```
@@ -710,7 +710,7 @@ persist_members_to_ring(State, ClusterName, Members) ->
 %% that were saved in the ring
 cluster_mgr_sites_fun() ->
     %% get cluster names from cluster manager
-    Ring = riak_core_ring_manager:get_my_ring(),
+    {ok, Ring} = riak_core_ring_manager:get_my_ring(),
     Clusters = riak_repl_ring:get_clusters(Ring),
     [{?CLUSTER_NAME_LOCATOR_TYPE, Name} || {Name, _Addrs} <- Clusters].

```

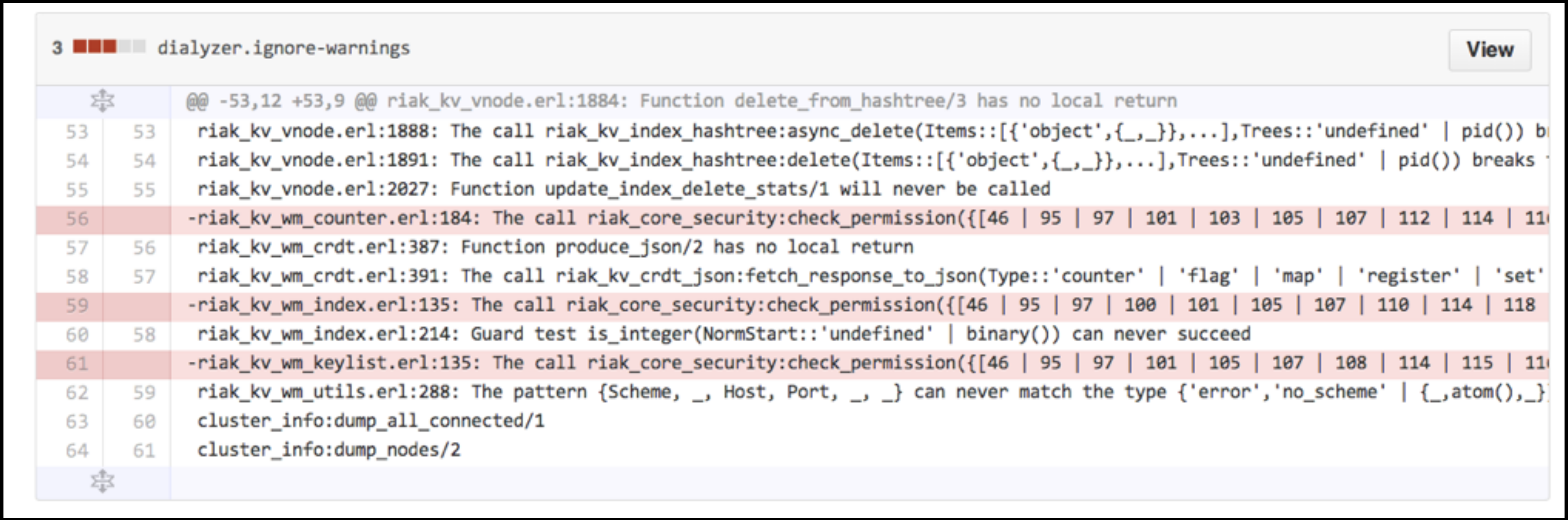
```erlang
%% @doc Return a list of all manifests in the
%% `active' or `writing' state
active_and_writing_manifests(Dict) ->
    orddict:to_list(filter_manifests_by_state(Dict,
                                              [active,
writing])).
```

```
%% @doc Return a list of all manifests in the
%% `active' or `writing' state
-spec active_and_writing_manifests(orddict:orddict()) ->
      [{binary(), lfs_manifest()}].
active_and_writing_manifests(Dict) ->
    orddict:to_list(filter_manifests_by_state(Dict,
                                              [active, writing])).
```

recently hooked it up with CI

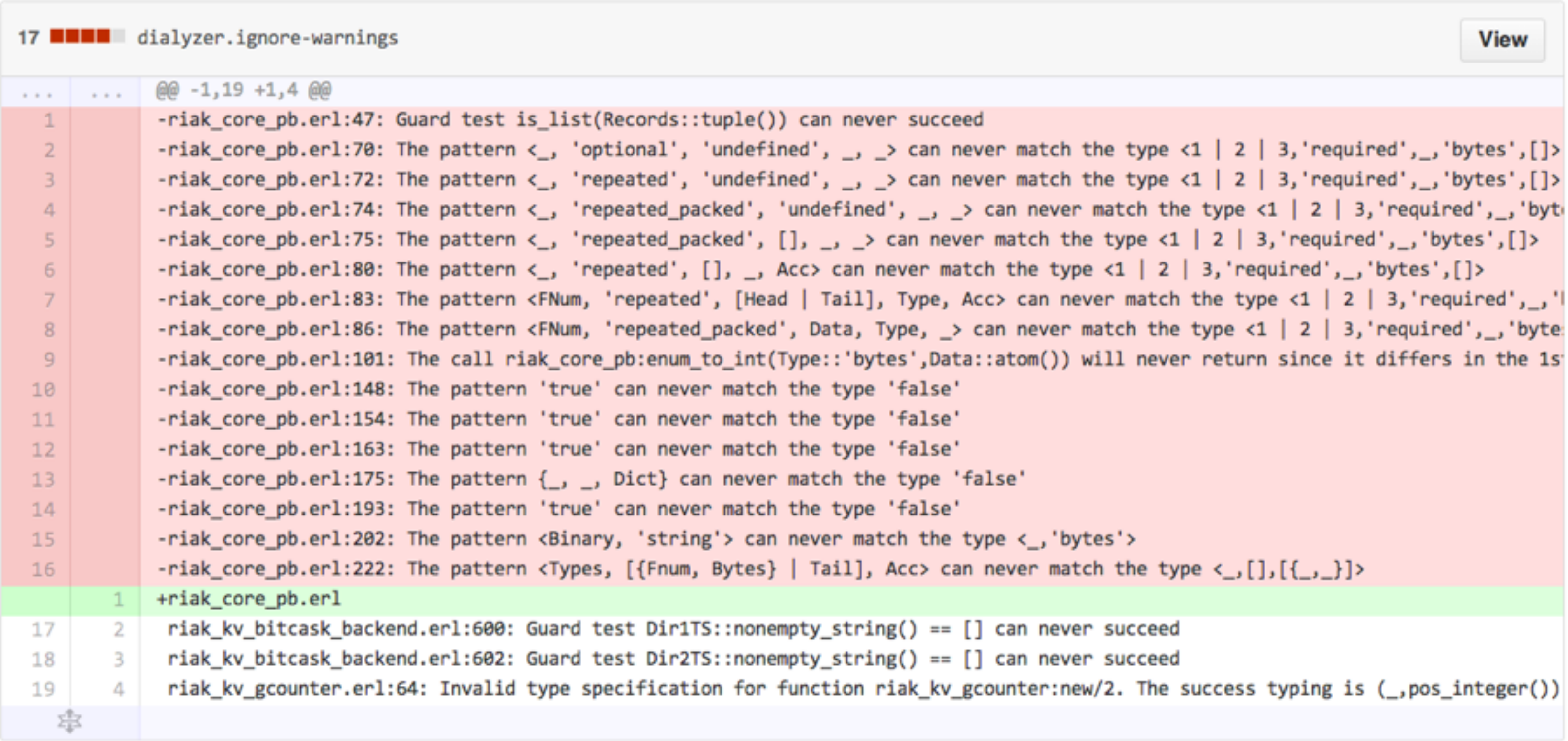


```
dialyzer $(DIALYZER_FLAGS) --plts $${PLTS} -c ebin > dialyzer_warnings ; \
cat dialyzer.ignore-warnings \
| sed -E 's/^([^:]+:)[^:]+:/\1/' \
| sort \
| uniq -c \
| sed -E '/.*\.erl: /!s/^[[:space:]]*[0-9]+[[:space:]]*//' \
> dialyzer.ignore-warnings.tmp ; \
egrep -v "^[[:space:]]*(done|Checking|Proceeding|Compiling)" dialyzer_warnings \
| sed -E 's/^([^:]+:)[^:]+:/\1/' \
| sort \
| uniq -c \
| sed -E '/.*\.erl: /!s/^[[:space:]]*[0-9]+[[:space:]]*//' \
| grep -F -f dialyzer.ignore-warnings.tmp -v \
| sed -E 's/^[[:space:]]*[0-9]+[[:space:]]*//' \
| sed -E 's/([]\^:+?|()*.$${}\[])/\\\1/g' \
| sed -E 's/(\\\.erl\\\:)/\1\\d+:/g' \
| sed -E 's/^(.*)$$/^\1$$/g' \
> dialyzer_unhandled_warnings ; \
rm dialyzer.ignore-warnings.tmp; \
if [ $$(cat dialyzer_unhandled_warnings | wc -l) -gt 0 ]; then \
    egrep -f dialyzer_unhandled_warnings dialyzer_warnings ; \
        found_warnings=1; \
```



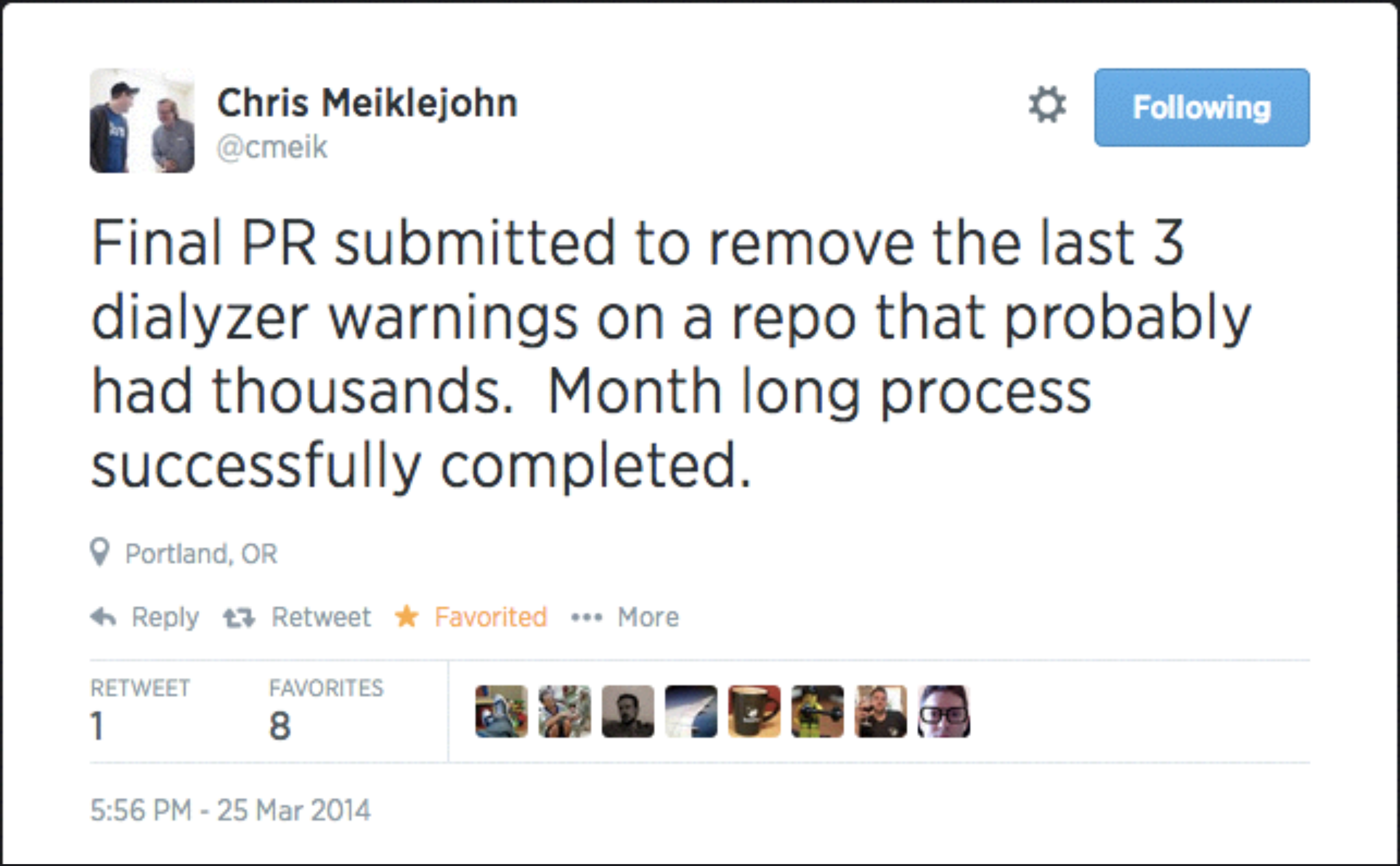
3 dialyzer.ignore-warnings

```
@@ -53,12 +53,9 @@ riak_kv_vnode.erl:1884: Function delete_from_hashtree/3 has no local return
 riak_kv_vnode.erl:1888: The call riak_kv_index_hashtree:async_delete(Items::[{'object',{_,_}},...],Trees::'undefined' | pid()) br
 riak_kv_vnode.erl:1891: The call riak_kv_index_hashtree:delete(Items::[{'object',{_,_}},...],Trees::'undefined' | pid()) breaks
 riak_kv_vnode.erl:2027: Function update_index_delete_stats/1 will never be called
-riak_kv_wm_counter.erl:184: The call riak_core_security:check_permission({[46 | 95 | 97 | 101 | 103 | 105 | 107 | 112 | 114 | 11
 riak_kv_wm_crdt.erl:387: Function produce_json/2 has no local return
 riak_kv_wm_crdt.erl:391: The call riak_kv_crdt_json:fetch_response_to_json(Type::'counter' | 'flag' | 'map' | 'register' | 'set'
-riak_kv_wm_index.erl:135: The call riak_core_security:check_permission({[46 | 95 | 97 | 100 | 101 | 105 | 107 | 110 | 114 | 118
 riak_kv_wm_index.erl:214: Guard test is_integer(NormStart::'undefined' | binary()) can never succeed
-riak_kv_wm_keylist.erl:135: The call riak_core_security:check_permission({[46 | 95 | 97 | 101 | 105 | 107 | 108 | 114 | 115 | 11
 riak_kv_wm_utils.erl:288: The pattern {Scheme, _, Host, Port, _, _} can never match the type {'error','no_scheme' | {_,atom(),_}
 cluster_info:dump_all_connected/1
 cluster_info:dump_nodes/2
```



17 dialyzer.ignore-warnings

```
@@ -1,19 +1,4 @@
-riak_core_pb.erl:47: Guard test is_list(Records::tuple()) can never succeed
-riak_core_pb.erl:70: The pattern <_, 'optional', 'undefined', _, _> can never match the type <1 | 2 | 3,'required',_,'bytes',[]>
-riak_core_pb.erl:72: The pattern <_, 'repeated', 'undefined', _, _> can never match the type <1 | 2 | 3,'required',_,'bytes',[]>
-riak_core_pb.erl:74: The pattern <_, 'repeated_packed', 'undefined', _, _> can never match the type <1 | 2 | 3,'required',_,'byt
-riak_core_pb.erl:75: The pattern <_, 'repeated_packed', [], _, _> can never match the type <1 | 2 | 3,'required',_,'bytes',[]>
-riak_core_pb.erl:80: The pattern <_, 'repeated', [], _, Acc> can never match the type <1 | 2 | 3,'required',_,'bytes',[]>
-riak_core_pb.erl:83: The pattern <FNum, 'repeated', [Head | Tail], Type, Acc> can never match the type <1 | 2 | 3,'required',_,'
-riak_core_pb.erl:86: The pattern <FNum, 'repeated_packed', Data, Type, _> can never match the type <1 | 2 | 3,'required',_,'byte
-riak_core_pb.erl:101: The call riak_core_pb:enum_to_int(Type::'bytes',Data::atom()) will never return since it differs in the 1s
-riak_core_pb.erl:148: The pattern 'true' can never match the type 'false'
-riak_core_pb.erl:154: The pattern 'true' can never match the type 'false'
-riak_core_pb.erl:163: The pattern 'true' can never match the type 'false'
-riak_core_pb.erl:175: The pattern {_, _, Dict} can never match the type 'false'
-riak_core_pb.erl:193: The pattern 'true' can never match the type 'false'
-riak_core_pb.erl:202: The pattern <Binary, 'string'> can never match the type <_,'bytes'>
-riak_core_pb.erl:222: The pattern <Types, [{Fnum, Bytes} | Tail], Acc> can never match the type <_,[],[{_,_}]>
+riak_core_pb.erl
 riak_kv_bitcask_backend.erl:600: Guard test Dir1TS::nonempty_string() == [] can never succeed
 riak_kv_bitcask_backend.erl:602: Guard test Dir2TS::nonempty_string() == [] can never succeed
 riak_kv_gcounter.erl:64: Invalid type specification for function riak_kv_gcounter:new/2. The success typing is (_,pos_integer())
```



Chris Meiklejohn
@cmeik

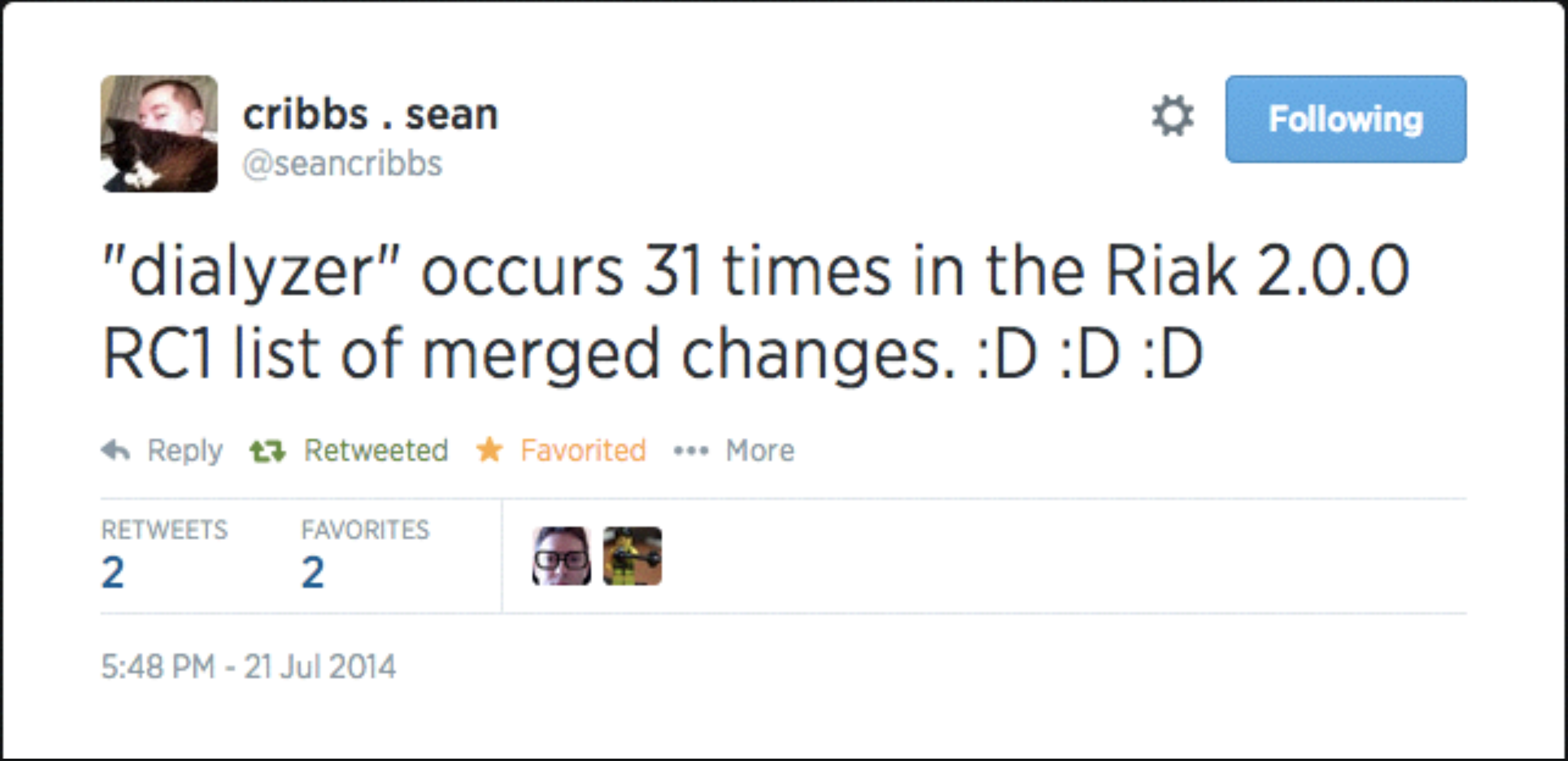
Final PR submitted to remove the last 3 dialyzer warnings on a repo that probably had thousands. Month long process successfully completed.

Portland, OR

RETWEET 1

FAVORITES 8

5:56 PM - 25 Mar 2014



cribbs . sean
@seancribbs

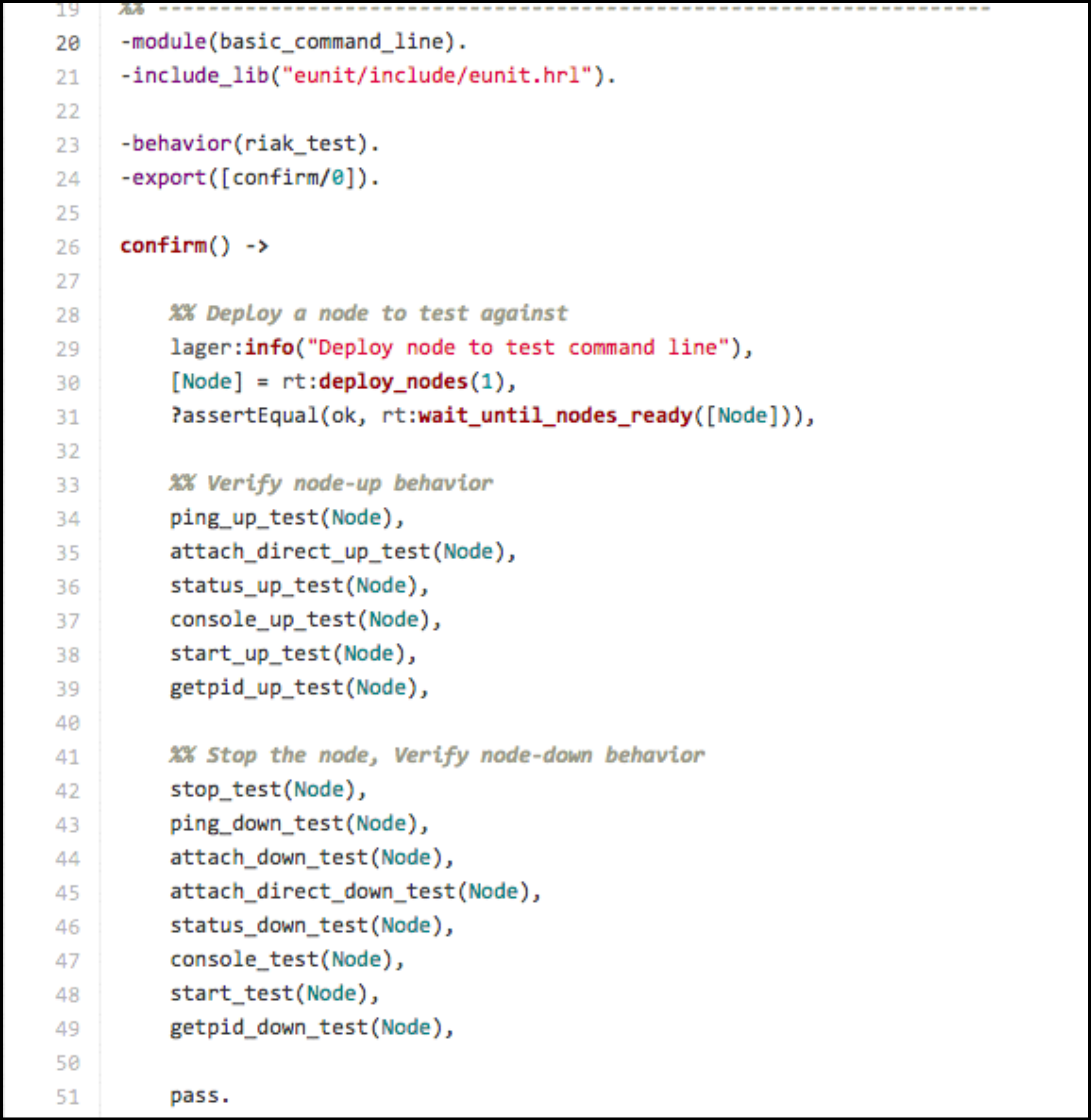
"dialyzer" occurs 31 times in the Riak 2.0.0 RC1 list of merged changes. :D :D :D

RETWEETS 2 FAVORITES 2

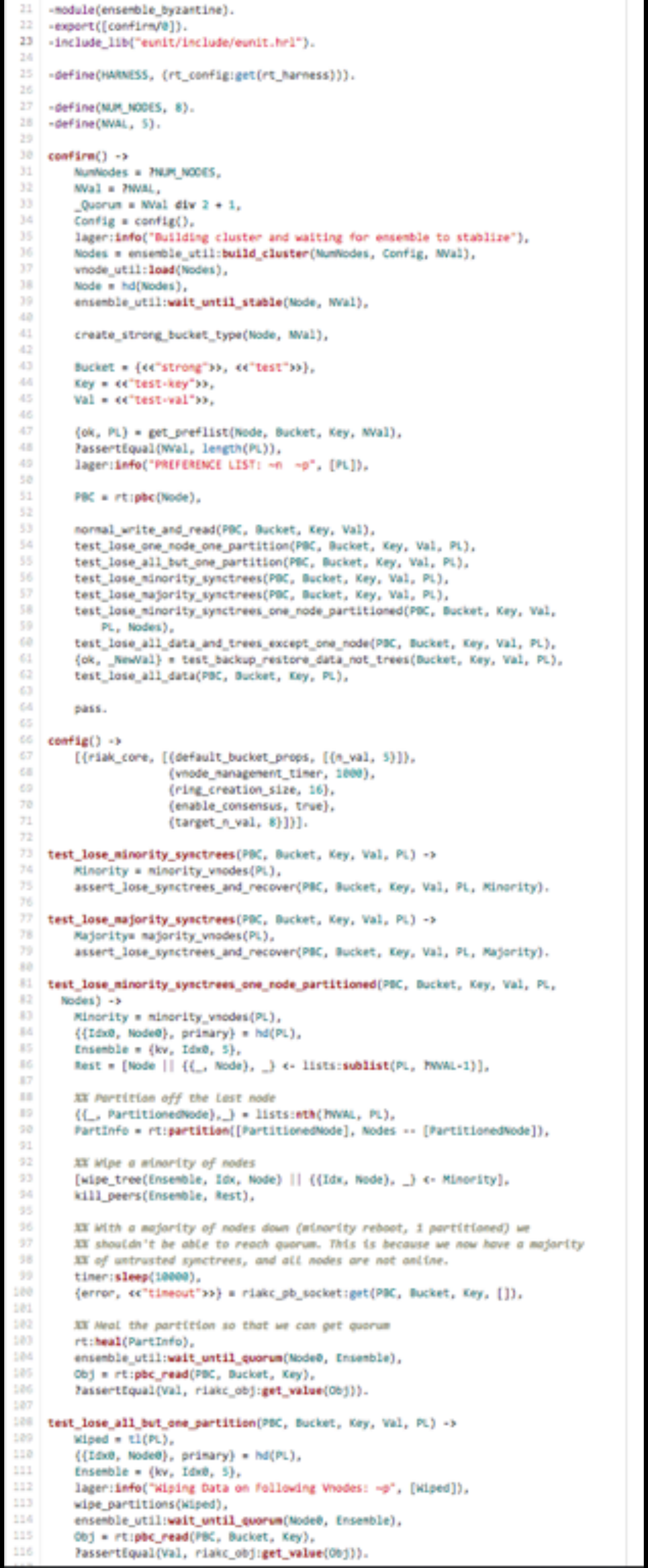
5:48 PM - 21 Jul 2014

riak-test

github.com/basho/
riak_test



```erlang
%% ------------------------------------------------------------------
-module(basic_command_line).
-include_lib("eunit/include/eunit.hrl").

-behavior(riak_test).
-export([confirm/0]).

confirm() ->

    %% Deploy a node to test against
    lager:info("Deploy node to test command line"),
    [Node] = rt:deploy_nodes(1),
    ?assertEqual(ok, rt:wait_until_nodes_ready([Node])),

    %% Verify node-up behavior
    ping_up_test(Node),
    attach_direct_up_test(Node),
    status_up_test(Node),
    console_up_test(Node),
    start_up_test(Node),
    getpid_up_test(Node),

    %% Stop the node, Verify node-down behavior
    stop_test(Node),
    ping_down_test(Node),
    attach_down_test(Node),
    attach_direct_down_test(Node),
    status_down_test(Node),
    console_test(Node),
    start_test(Node),
    getpid_down_test(Node),

    pass.
```



```
-module(ensemble_byzantine).
-export([confirm/0]).
-include_lib("eunit/include/eunit.hrl").

-define(HARNESS, (rt_config:get(rt_harness))).

-define(NUM_NODES, 8).
-define(NVAL, 5).

confirm() ->
    NumNodes = ?NUM_NODES,
    NVal = ?NVAL,
    _Quorum = NVal div 2 + 1,
    Config = config(),
    lager:info("Building cluster and waiting for ensemble to stablize"),
    Nodes = ensemble_util:build_cluster(NumNodes, Config, NVal),
    vnode_util:load(Nodes),
    Node = hd(Nodes),
    ensemble_util:wait_until_stable(Node, NVal),

    create_strong_bucket_type(Node, NVal),

    Bucket = {<<"strong">>, <<"test">>},
    Key = <<"test-key">>,
    Val = <<"test-val">>,

    {ok, PL} = get_preflist(Node, Bucket, Key, NVal),
    ?assertEqual(NVal, length(PL)),
    lager:info("PREFERENCE LIST: ~n  ~p", [PL]),

    PBC = rt:pbc(Node),

    normal_write_and_read(PBC, Bucket, Key, Val),
    test_lose_one_node_one_partition(PBC, Bucket, Key, Val, PL),
    test_lose_all_but_one_partition(PBC, Bucket, Key, Val, PL),
    test_lose_minority_synctrees(PBC, Bucket, Key, Val, PL),
    test_lose_majority_synctrees(PBC, Bucket, Key, Val, PL),
    test_lose_minority_synctrees_one_node_partitioned(PBC, Bucket, Key, Val,
        PL, Nodes),
    test_lose_all_data_and_trees_except_one_node(PBC, Bucket, Key, Val, PL),
    {ok, _NewVal} = test_backup_restore_data_not_trees(Bucket, Key, Val, PL),
    test_lose_all_data(PBC, Bucket, Key, PL),

    pass.

config() ->
    [{riak_core, [{default_bucket_props, [{n_val, 5}]},
                  {vnode_management_timer, 1000},
                  {ring_creation_size, 16},
                  {enable_consensus, true},
                  {target_n_val, 8}]}].

test_lose_minority_synctrees(PBC, Bucket, Key, Val, PL) ->
    Minority = minority_vnodes(PL),
    assert_lose_synctrees_and_recover(PBC, Bucket, Key, Val, PL, Minority).

test_lose_majority_synctrees(PBC, Bucket, Key, Val, PL) ->
    Majority= majority_vnodes(PL),
    assert_lose_synctrees_and_recover(PBC, Bucket, Key, Val, PL, Majority).

test_lose_minority_synctrees_one_node_partitioned(PBC, Bucket, Key, Val, PL,
  Nodes) ->
    Minority = minority_vnodes(PL),
    {{Idx0, Node0}, primary} = hd(PL),
    Ensemble = {kv, Idx0, 5},
    Rest = [Node || {{_, Node}, _} <- lists:sublist(PL, ?NVAL-1)],

    %% Partition off the last node
    {{_, PartitionedNode},_} = lists:nth(?NVAL, PL),
    PartInfo = rt:partition([PartitionedNode], Nodes -- [PartitionedNode]),

    %% Wipe a minority of nodes
    [wipe_tree(Ensemble, Idx, Node) || {{Idx, Node}, _} <- Minority],
    kill_peers(Ensemble, Rest),

    %% With a majority of nodes down (minority reboot, 1 partitioned) we
    %% shouldn't be able to reach quorum. This is because we now have a majority
    %% of untrusted synctrees, and all nodes are not online.
    timer:sleep(10000),
    {error, <<"timeout">>} = riakc_pb_socket:get(PBC, Bucket, Key, []),

    %% Heal the partition so that we can get quorum
    rt:heal(PartInfo),
    ensemble_util:wait_until_quorum(Node0, Ensemble),
    Obj = rt:pbc_read(PBC, Bucket, Key),
    ?assertEqual(Val, riakc_obj:get_value(Obj)).

test_lose_all_but_one_partition(PBC, Bucket, Key, Val, PL) ->
    Wiped = tl(PL),
    {{Idx0, Node0}, primary} = hd(PL),
    Ensemble = {kv, Idx0, 5},
    lager:info("Wiping Data on Following Vnodes: ~p", [Wiped]),
    wipe_partitions(Wiped),
    ensemble_util:wait_until_quorum(Node0, Ensemble),
    Obj = rt:pbc_read(PBC, Bucket, Key),
    ?assertEqual(Val, riakc_obj:get_value(Obj)).
```

EDoc

tools.mk

github.com/basho/
tools.mk

A small makefile library for working with Erlang tools

compile-no-deps test
upload-docs docs xref
plt local_plt dialyzer
dialyzer-quick cleanplt

Erlang QuickCheck

(EQC)

EQC is a secret weapon

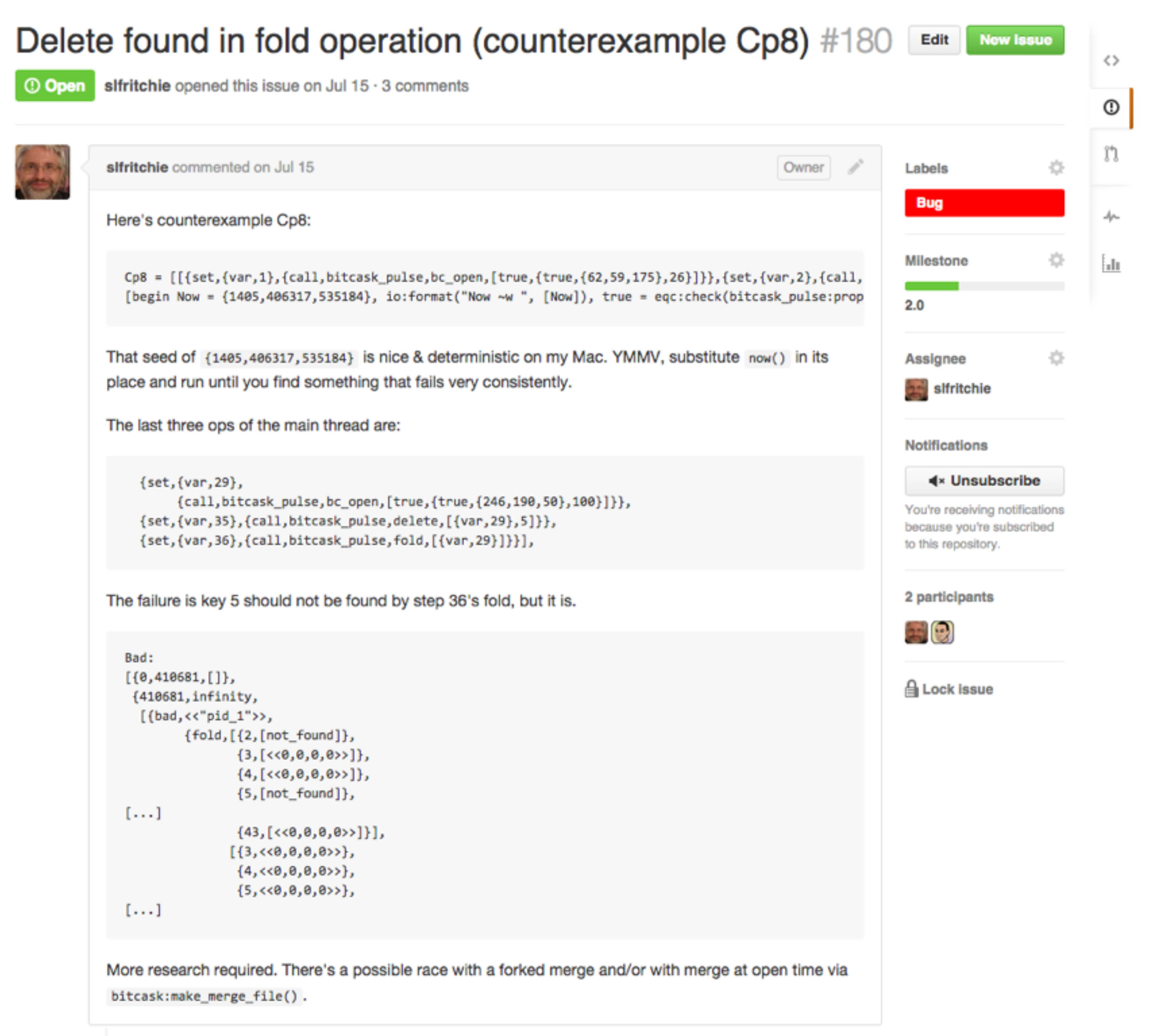
quviq.com

triq PropEr

for all lists L:

```
reverse(reverse(L)) =:= L.
```

For all get/put/delete operations a key/value store should mimic a hash-map



Delete found in fold operation (counterexample Cp8) #180

slfritchie opened this issue on Jul 15 · 3 comments

slfritchie commented on Jul 15

Here's counterexample Cp8:

```
Cp8 = [[{set,{var,1},{call,bitcask_pulse,bc_open,[true,{true,{62,59,175},26}]}},{set,{var,2},{call,
[begin Now = {1405,406317,535184}, io:format("Now ~w ", [Now]), true = eqc:check(bitcask_pulse:prop
```

That seed of `{1405,406317,535184}` is nice & deterministic on my Mac. YMMV, substitute `now()` in its place and run until you find something that fails very consistently.

The last three ops of the main thread are:

```
  {set,{var,29},
       {call,bitcask_pulse,bc_open,[true,{true,{246,190,50},100}]}},
  {set,{var,35},{call,bitcask_pulse,delete,[{var,29},5]}},
  {set,{var,36},{call,bitcask_pulse,fold,[{var,29}]}}],
```

The failure is key 5 should not be found by step 36's fold, but it is.

```
Bad:
[{0,410681,[]},
 {410681,infinity,
  [{bad,<<"pid_1">>,
        {fold,[{2,[not_found]},
               {3,[<<0,0,0,0>>]},
               {4,[<<0,0,0,0>>]},
               {5,[not_found]},
[...]
               {43,[<<0,0,0,0>>]}],
              [{3,<<0,0,0,0>>},
               {4,<<0,0,0,0>>},
               {5,<<0,0,0,0>>},
[...]
```

More research required. There's a possible race with a forked merge and/or with merge at open time via `bitcask:make_merge_file()`.

http://bit.ly/1rDOGr3

poolboy race conditions

basho.com/quickchecking-poolboy-for-fun-and-profit/

"Now, poolboy had 85% test coverage (and most of the remaining 15% was irrelevant boilerplate) when I started QuickChecking it, and I felt pretty happy with its solidity, so I didn’t expect to find many bugs, if any. I was very wrong."

Andrew Thompson

Questions?

Reid Draper @reiddraper