# The MySQL Ecosystem @ GitHub

# Sam Lambert

**Director of Technology**

github.com/samlambert
twitter.com/isamlambert
samlambert.com

*THE HUMAN BRAIN IS A WONDERFUL THING. IT STARTS WORKING THE MOMENT YOU ARE BORN, AND NEVER STOPS UNTIL YOU STAND UP TO SPEAK IN PUBLIC.*

- GEORGE JESSEL

# GitHub

**Code hosting and collaboration**

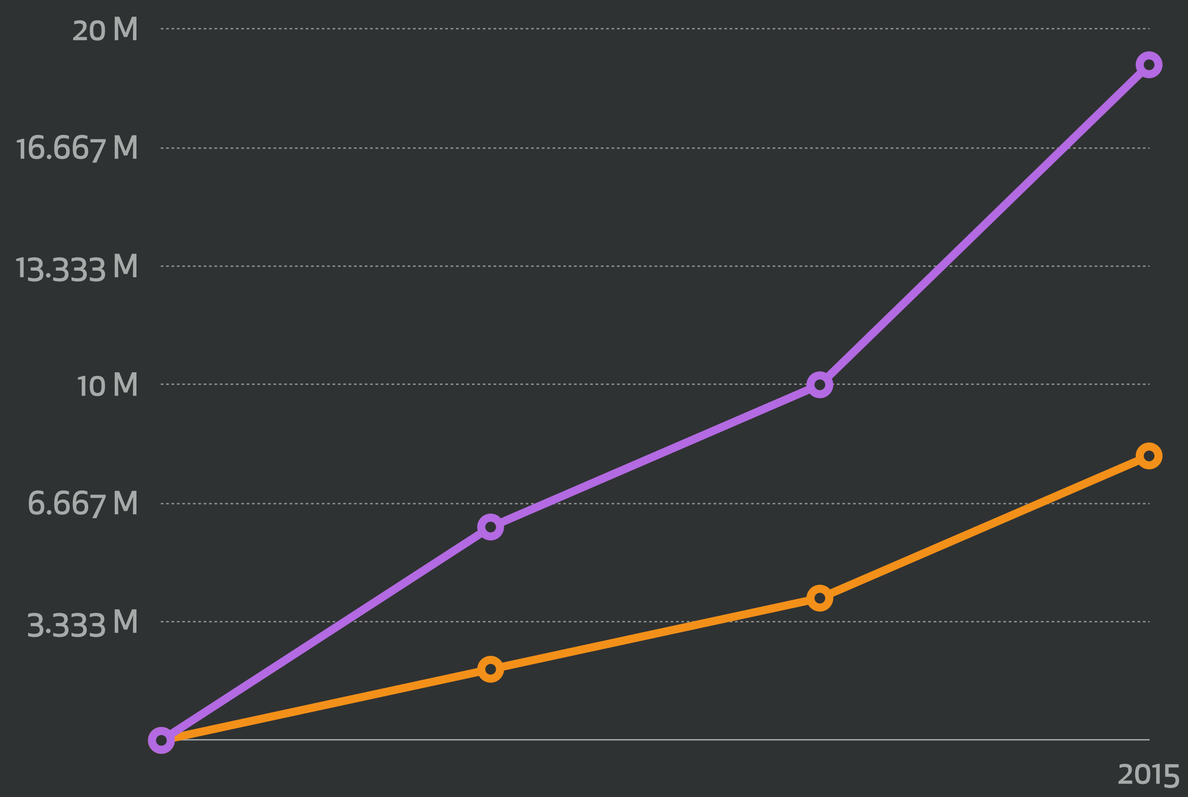
# What is GitHub?

**8**
**MILLION**

DEVELOPERS ARE WORKING TOGETHER ON

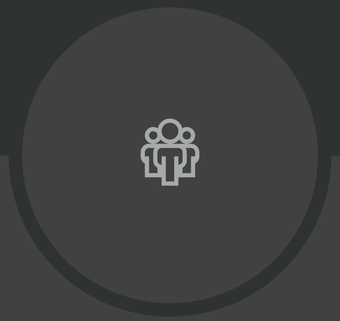
**21**
**MILLION**

PROJECTS ON GITHUB RIGHT NOW



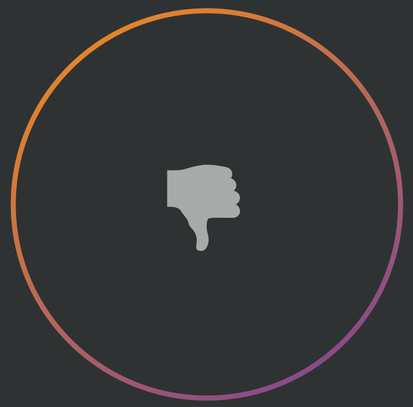
# What is GitHub?

130+ TB
**Git data**

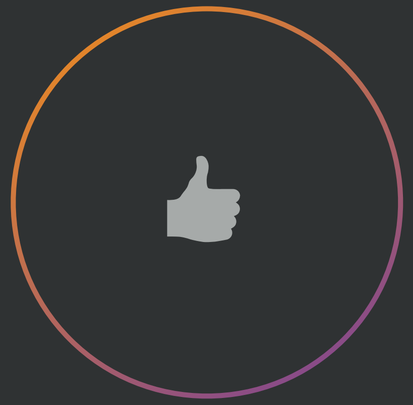


239
**GitHubbers**

100
**Engineers**



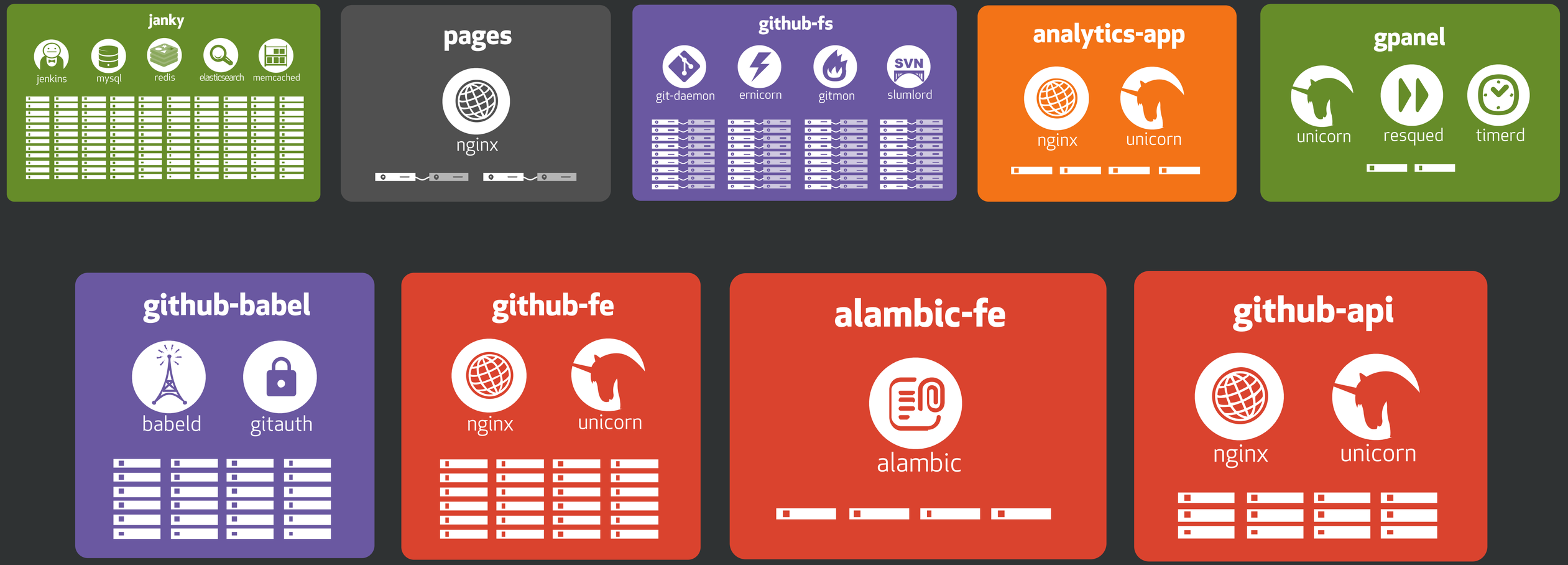
# Slave



# Replica/Secondary

# What is MySQL's role at GitHub?

# What is MySQL's role at GitHub?



# What is the Ecosystem?

# What is the Ecosystem?

# Why is the Ecosystem so important?

# MySQL is fine on its own

# We can (mostly) trust MySQL

# The GitHub Stack

# The GitHub Stack

## WEB

### Ruby on Rails
Powerful application framework.
Rapid development.

### Javascript
UI goodness

## BACKEND

### C
Powers a number of backend applications

### Puppet
Provisioning

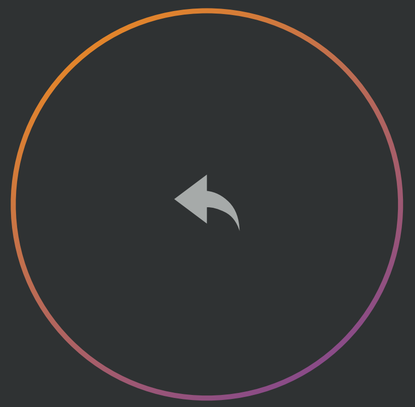
### Git
Obviously

## DATABASES

### MySQL
Main source of truth

### Elasticsearch
Indexes all the code, as well as issues and pull requests.

### Redis
Resque, cache data.



```
SELECT DATE_SUB(NOW(),
INTERVAL 2 YEAR);
```

# Small number of MySQL hosts

# Majority of queries served from one host

# Replicas only used for backups and failover

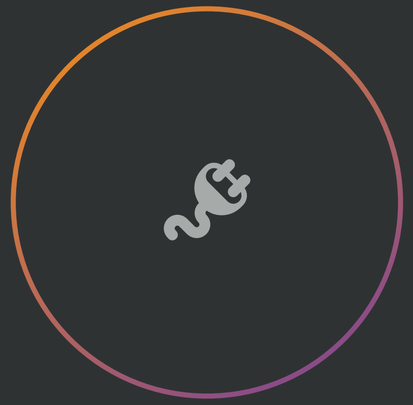
# Old hardware

# Unscalable

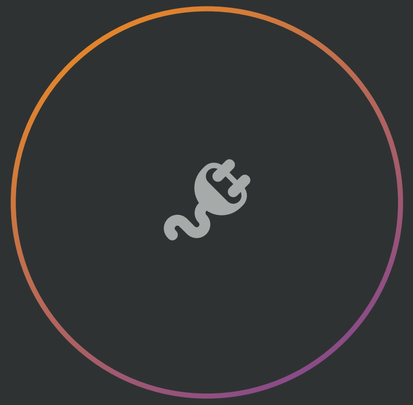
# Contention everywhere

# Traffic spikes caused query response times to go up

# Time for a change!



# Need to move data centers



# A chance to update hardware

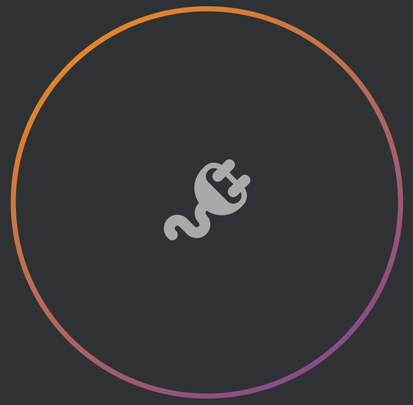
# New Datacenter

## HARDWARE

- More hosts
- Faster CPUs
- SSDs

## NETWORKING

- 10gb



# A chance to tune config

# Time to functionally shard

# Functional Sharding

- Partition by function
- Split features out
- Isolate tables as a whole
- App specific
- Milage may vary

# Large volume single table

# Constantly growing

# No joins - app refactor

# Regression testing is essential

# Replaying queries from live

# Long benchmarks +4 hours

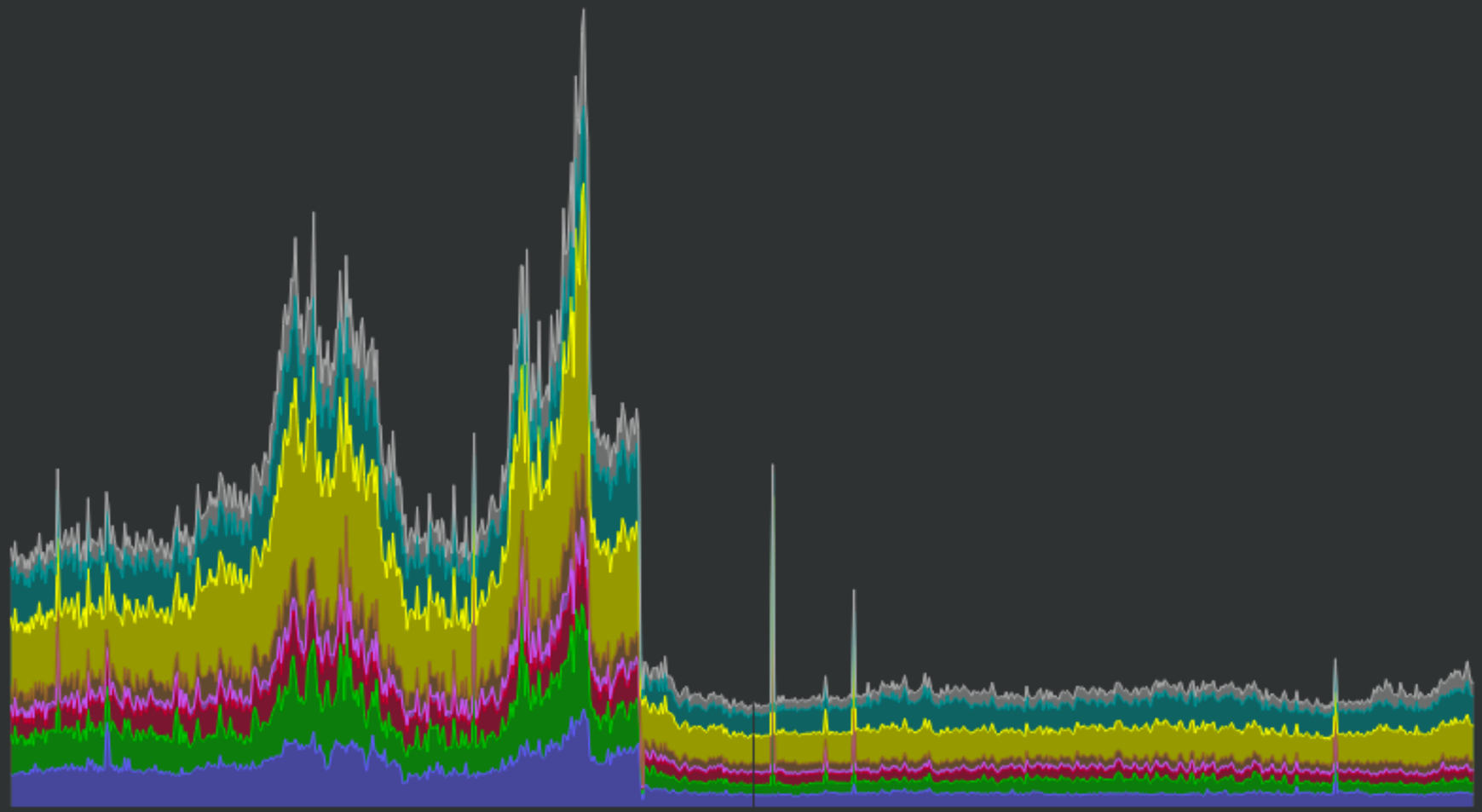
# Go Live

# Maintenance window

13 minutes

# Results!

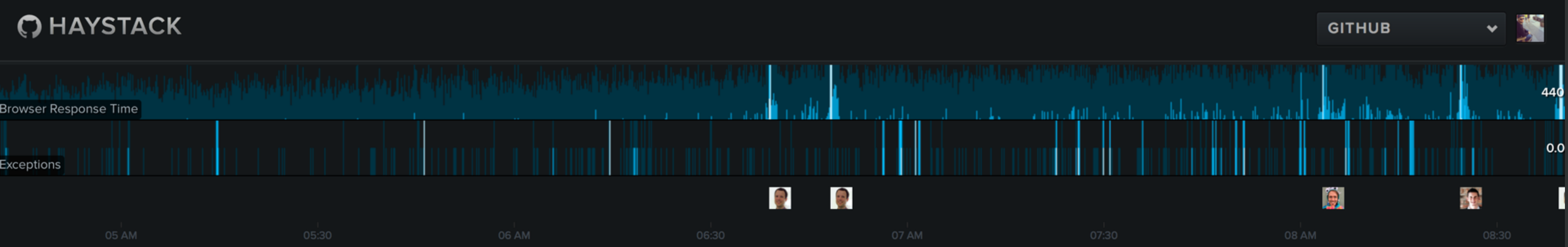
## MySQL app mean time



# Tooling

# Lots to mention

# Haystack




Search...

Overview

Dashboard

Firehose

Experienced by you

Latest Deploys

**flipper-stafftools**
deployed by aroben a minute ago

**billing-manager-emails**
deployed by shayfrendt 16 minutes ago

**lfs-data-quotas-refactor**
deployed by tclem 37 minutes ago

**flipper-peek-bar**
deployed by aroben 2 hours ago

**flipper-middleware**
deployed by aroben 2 hours ago

# Dashboard

Updated: April 13, 2015 08:40:59

5m | 30m | 1h | 3h | 1d

**GitHub::TimeoutMiddleware::RequestTimeout** github/github#30415 github/github#31070 github/github#39550 github/github#31069 — 3
last occurred 2 minutes ago — users#show

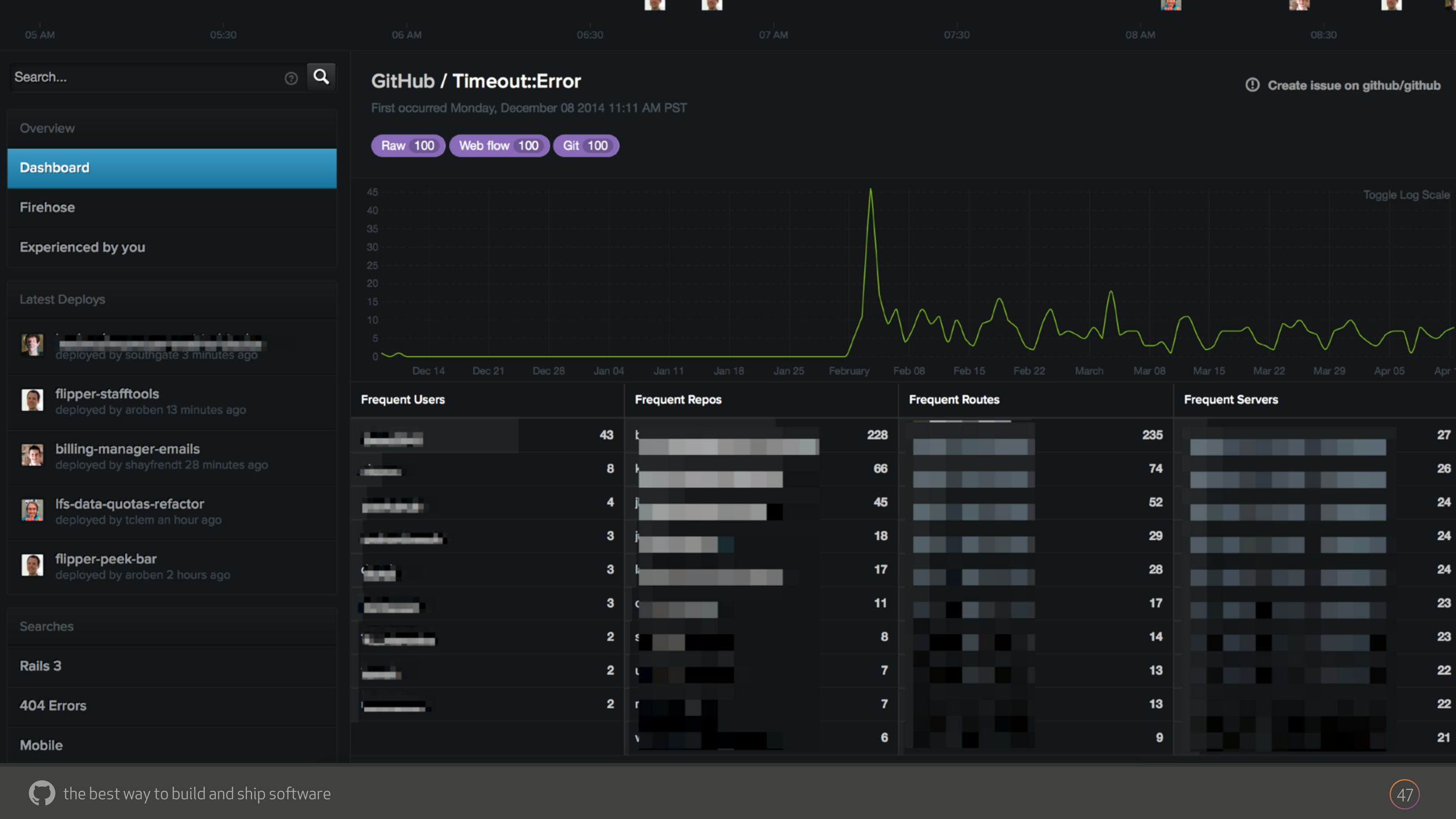
**NoMethodError** — 2
last occurred 3 minutes ago — undefined method `repository' for nil:NilClass

**Braintree::NotFoundError** — 1
last occurred 2 minutes ago

**GitRPC::Timeout** — 1
last occurred 4 minutes ago — GitRPC::Timeout

**GitHub::TimeoutMiddleware::RequestTimeout** — 1
last occurred just now — blob#show

**Timeout::Error** — 1
last occurred 2 minutes ago — execution expired


05 AM
05:30
06 AM
06:30
07 AM
07:30
08 AM
08:30
Search...
Overview
Dashboard
Firehose
Experienced by you
Latest Deploys
deployed by southgate 3 minutes ago
flipper-stafftools
deployed by aroben 13 minutes ago
billing-manager-emails
deployed by shayfrendt 28 minutes ago
lfs-data-quotas-refactor
deployed by tclem an hour ago
flipper-peek-bar
deployed by aroben 2 hours ago
Searches
Rails 3
404 Errors
Mobile
GitHub / Timeout::Error
First occurred Monday, December 08 2014 11:11 AM PST
Create issue on github/github
Raw 100
Web flow 100
Git 100
Toggle Log Scale
45
40
35
30
25
20
15
10
5
0
Dec 14
Dec 21
Dec 28
Jan 04
Jan 11
Jan 18
Jan 25
February
Feb 08
Feb 15
Feb 22
March
Mar 08
Mar 15
Mar 22
Mar 29
Apr 05
Frequent Users
43
8
4
3
3
3
2
2
2
Frequent Repos
228
66
45
18
17
11
8
7
7
6
Frequent Routes
235
74
52
29
28
17
14
13
13
9
Frequent Servers
27
26
24
24
24
23
23
22
22
21

Search...

Overview

Dashboard

Firehose

Experienced by you

Latest Deploys

[redacted]
deployed by southgate 5 minutes ago

flipper-stafftools
deployed by aroben 15 minutes ago

billing-manager-emails
deployed by shayfrendt 30 minutes ago

lfs-data-quotas-refactor
deployed by tclem an hour ago

flipper-peek-bar
deployed by aroben 2 hours ago

Searches

Rails 3

404 Errors

Mobile

User Errors

Machine Room

# GitHub / Timeout::Error

April 13, 2015 08:34:58

Create issue on github/github | Related needles

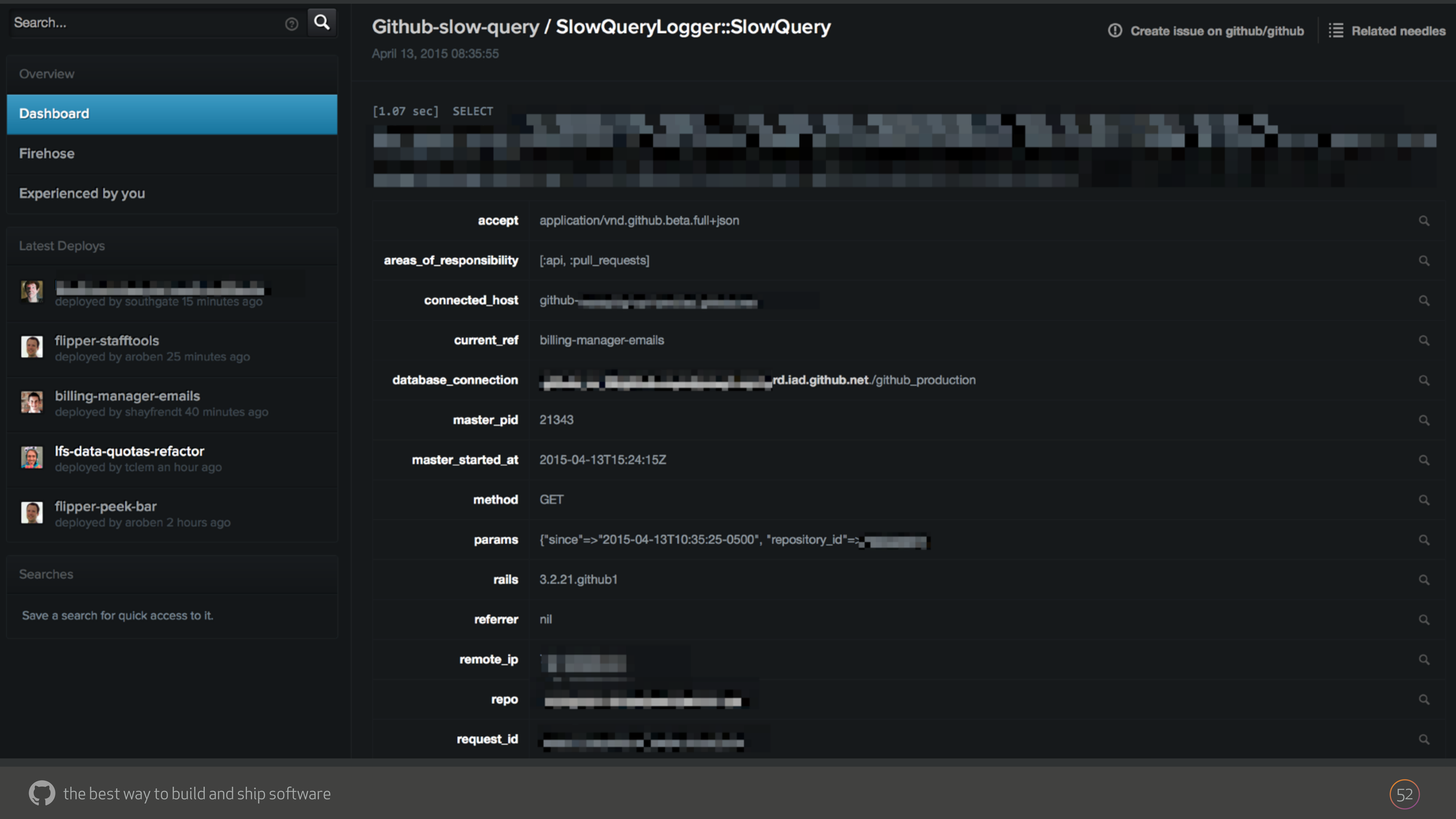
```
execution expired
```

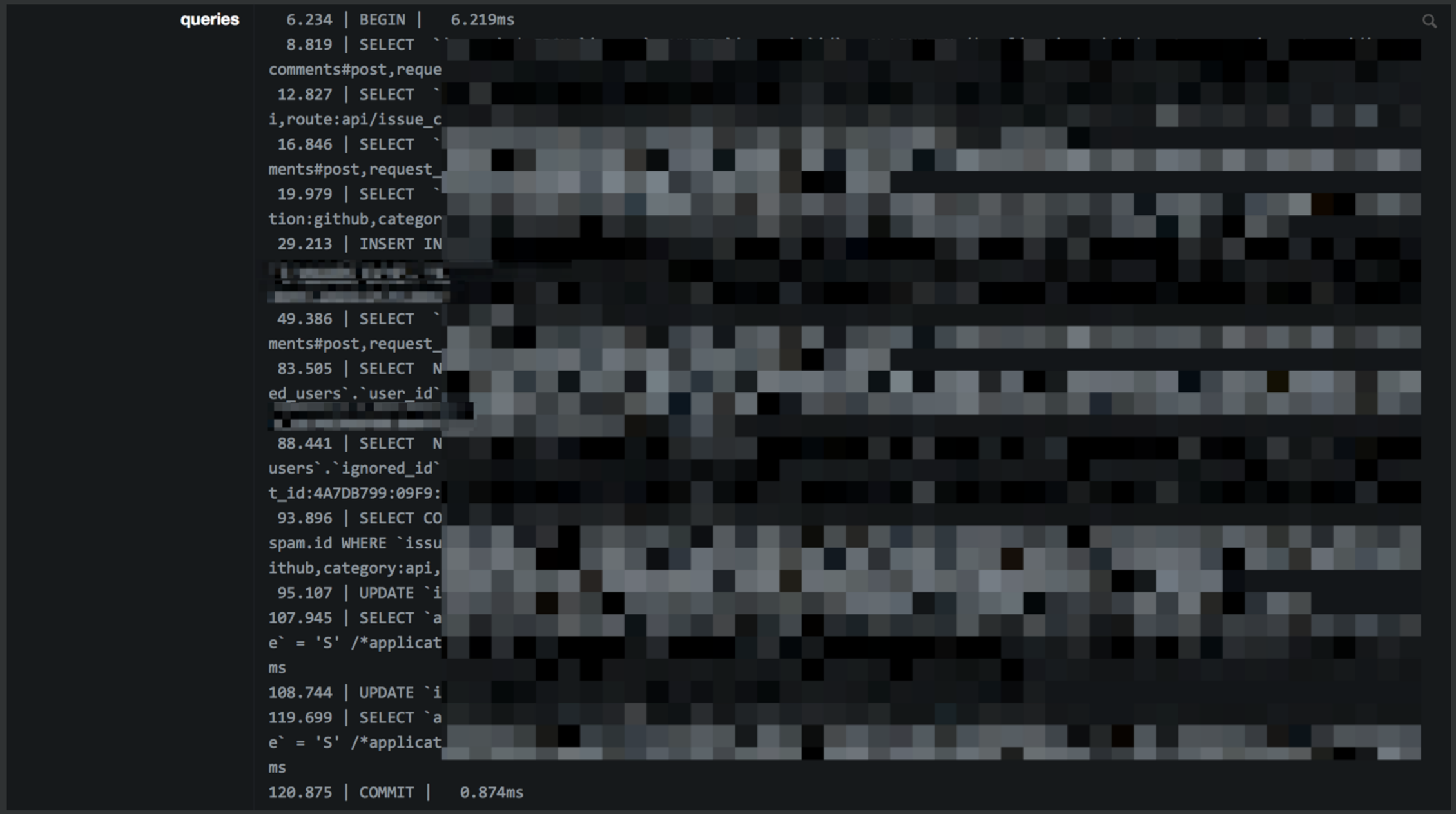
| | |
|---|---|
| **action** | BlobController#show |
| **accept** | text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 |
| **areas_of_responsibility** | [:raw, :web_flow, :git] |
| **current_ref** | billing-manager-emails |
| **dbconn** | write |
| **master_pid** | 21527 |
| **master_started_at** | 2015-04-13T15:24:06Z |
| **method** | GET |
| **newsies_dbconn** | notifications_primary |
| **oauth_access_id** | nil |
| **params** | {"controller"=>"blob", "action"=>"show", "user_id"=>"[redacted]" |
| **rails** | 3.2.21.github1 |
| **referrer** | https://github.com/[redacted] |
| **remote_ip** | [redacted] |
| **repo** | [redacted] |
| **request_id** | [redacted] |

```
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/xpath_context.rb:8:in `register_namespaces'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node.rb:152:in `block in xpath'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node.rb:150:in `map'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node.rb:150:in `xpath'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node_set.rb:114:in `block in css'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node_set.rb:237:in `block in each'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node_set.rb:236:in `upto'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node_set.rb:236:in `each'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/node_set.rb:103:in `css'
vendor/gems/2.1.5/ruby/2.1.0/gems/nokogiri-1.6.3.1/lib/nokogiri/xml/document_fragment.rb:79:in `css'
lib/github/html/name_prefix_filter.rb:30:in `call'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline/filter.rb:137:in `call'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:121:in `block in perform_filter'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:161:in `block in instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications.rb:123:in `block in instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications/instrumenter.rb:20:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications.rb:123:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:160:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:120:in `perform_filter'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:106:in `block (2 levels) in call'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:105:in `each'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:105:in `inject'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:105:in `block in call'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:161:in `block in instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications.rb:123:in `block in instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications/instrumenter.rb:20:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/activesupport-3.2.21.github1/lib/active_support/notifications.rb:123:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:160:in `instrument'
vendor/gems/2.1.5/ruby/2.1.0/gems/html-pipeline-1.11.0/lib/html/pipeline.rb:103:in `call'
app/helpers/blob_markup_helper.rb:116:in `block in markup_blob!'
vendor/ruby/71c83aaea957a8b31cf30ecf1e622414c7370815/lib/ruby/2.1.0/timeout.rb:91:in `block in timeout'
vendor/ruby/71c83aaea957a8b31cf30ecf1e622414c7370815/lib/ruby/2.1.0/timeout.rb:35:in `block in catch'
vendor/ruby/71c83aaea957a8b31cf30ecf1e622414c7370815/lib/ruby/2.1.0/timeout.rb:35:in `catch'
vendor/ruby/71c83aaea957a8b31cf30ecf1e622414c7370815/lib/ruby/2.1.0/timeout.rb:35:in `catch'
vendor/ruby/71c83aaea957a8b31cf30ecf1e622414c7370815/lib/ruby/2.1.0/timeout.rb:106:in `timeout'
app/helpers/blob_markup_helper.rb:115:in `markup_blob!'
app/helpers/blob_markup_helper.rb:97:in `markup_blob'
app/helpers/blob_markup_helper.rb:36:in `format_blob'
app/views/blob/_blob_content.html.erb:1:in `_app_views_blob__blob_content_html_erb__757533915030191661_69855271779560'
```

```
27      def call
28        return doc unless prefix
29
30        doc.css('*[name], *[id]').each do |element|
31          %w(name id).each do |attribute|
32            if element[attribute]
33              element[attribute] = element[attribute].sub(/\A(#{prefix})?/, prefix)
34            end
35          end
36        end
```

# Slow queries


Search...
Overview
Dashboard
Firehose
Experienced by you
Latest Deploys
deployed by southgate 15 minutes ago
flipper-stafftools
deployed by aroben 25 minutes ago
billing-manager-emails
deployed by shayfrendt 40 minutes ago
lfs-data-quotas-refactor
deployed by tclem an hour ago
flipper-peek-bar
deployed by aroben 2 hours ago
Searches
Save a search for quick access to it.
Github-slow-query / SlowQueryLogger::SlowQuery
April 13, 2015 08:35:55
Create issue on github/github
Related needles
[1.07 sec] SELECT
accept
application/vnd.github.beta.full+json
areas_of_responsibility
[:api, :pull_requests]
connected_host
github-
current_ref
billing-manager-emails
database_connection
rd.iad.github.net./github_production
master_pid
21343
master_started_at
2015-04-13T15:24:15Z
method
GET
params
{"since"=>"2015-04-13T10:35:25-0500", "repository_id"=>
rails
3.2.21.github1
referrer
nil
remote_ip
repo
request_id


queries
6.234 | BEGIN | 6.219ms
8.819 | SELECT `
comments#post,reque
12.827 | SELECT `
i,route:api/issue_c
16.846 | SELECT `
ments#post,request_
19.979 | SELECT `
tion:github,categor
29.213 | INSERT IN
49.386 | SELECT `
ments#post,request_
83.505 | SELECT N
ed_users`.`user_id`
88.441 | SELECT N
users`.`ignored_id`
t_id:4A7DB799:09F9:
93.896 | SELECT CO
spam.id WHERE `issu
ithub,category:api,
95.107 | UPDATE `i
107.945 | SELECT `a
e` = 'S' /*applicat
ms
108.744 | UPDATE `i
119.699 | SELECT `a
e` = 'S' /*applicat
ms
120.875 | COMMIT | 0.874ms

# Haystack is awesome

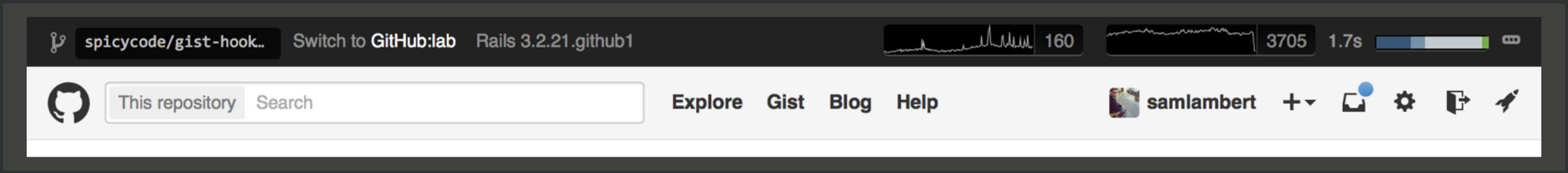
# Failbotd

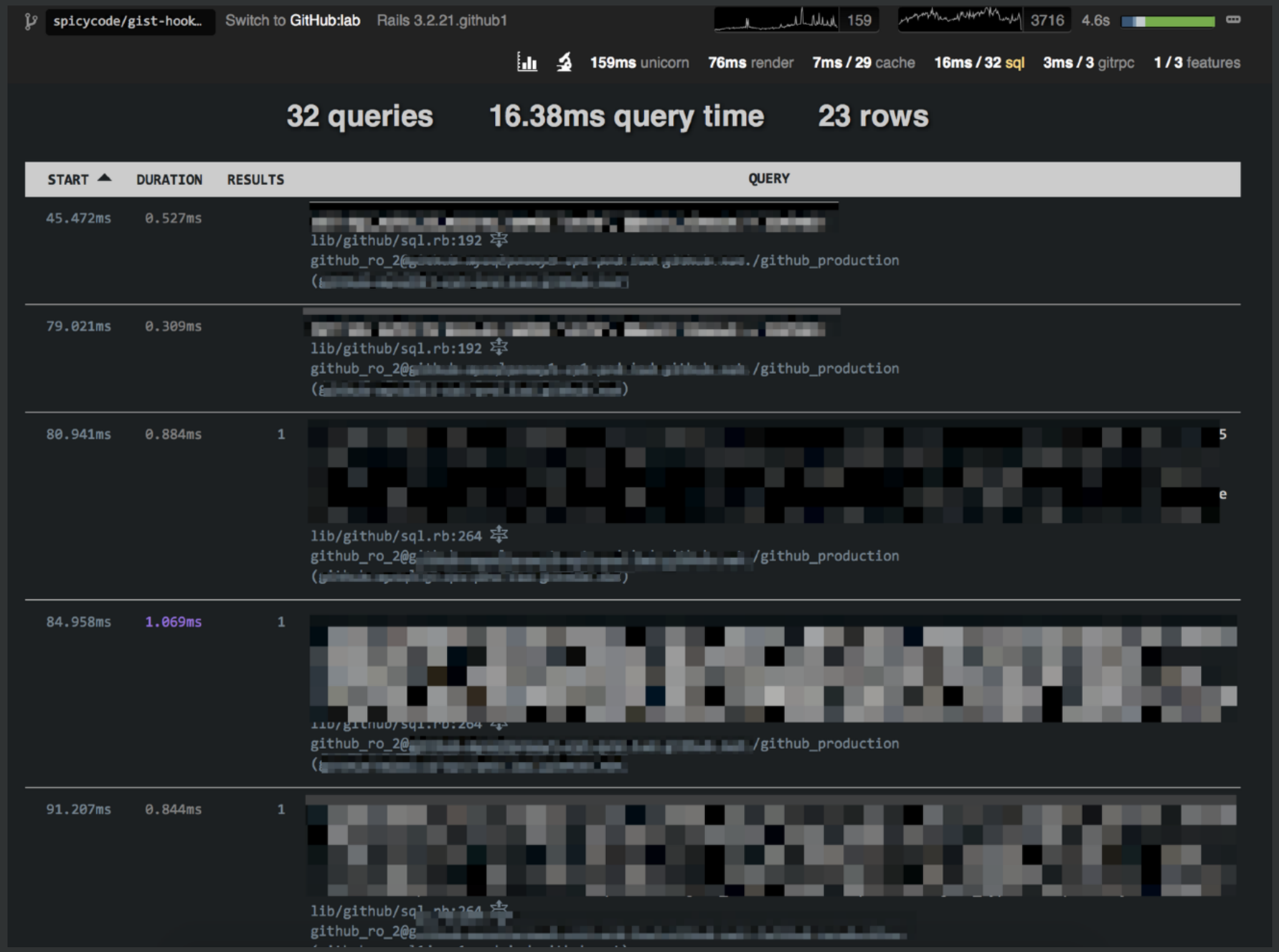
# Failbotd

- C daemon
- Takes exceptions from the app
- Queues them
- Posts to Haystack

# Staff toolbar

# Staff toolbar



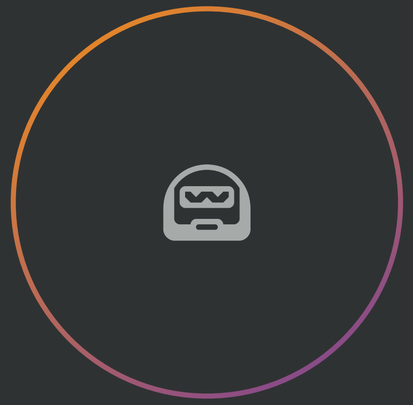

spicycode/gist-hook..
Switch to GitHub:lab
Rails 3.2.21.github1
159
3716
4.6s
159ms unicorn
76ms render
7ms / 29 cache
16ms / 32 sql
3ms / 3 gitrpc
1 / 3 features
32 queries
16.38ms query time
23 rows
START
DURATION
RESULTS
QUERY
45.472ms
0.527ms
lib/github/sql.rb:192
github_ro_2@
./github_production
79.021ms
0.309ms
lib/github/sql.rb:192
github_ro_2@
/github_production
80.941ms
0.884ms
1
lib/github/sql.rb:264
github_ro_2@
/github_production
84.958ms
1.069ms
1
github_ro_2@
/github_production
91.207ms
0.844ms
1
lib/github/sql.rb:264
github_ro_2@

# Chat

~137 remote employees

# 100+ chat rooms

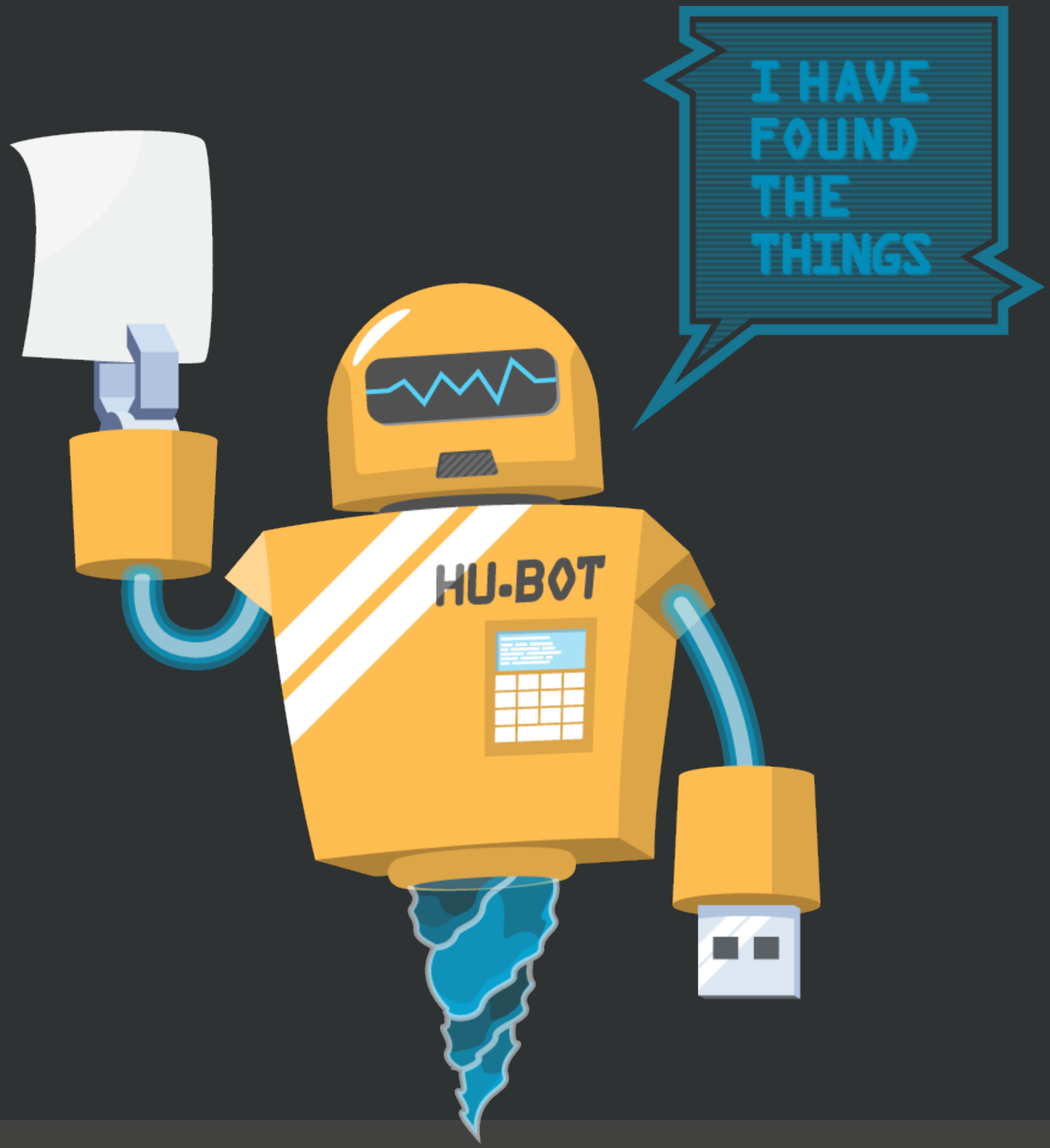
# Hubot



# Who is Hubot?

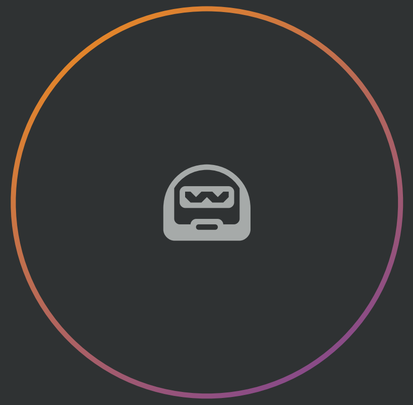
# Who is Hubot?

*GITHUB, INC., WROTE THE FIRST VERSION OF HUBOT TO AUTOMATE OUR COMPANY CHAT ROOM. HUBOT KNEW HOW TO DEPLOY THE SITE, AUTOMATE A LOT OF TASKS, AND BE A SOURCE OF FUN IN THE COMPANY. EVENTUALLY HE GREW TO BECOME A FORMIDABLE FORCE IN GITHUB. BUT HE LED A PRIVATE, MESSY LIFE. SO WE REWROTE HIM.*

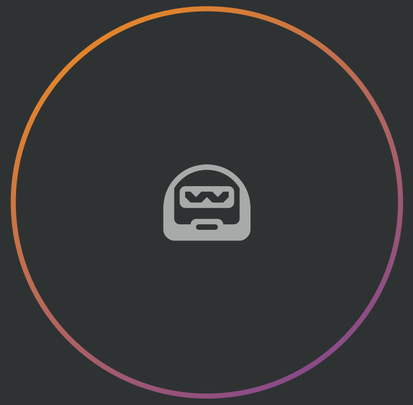

I HAVE FOUND THE THINGS
HU-BOT

# Open Source

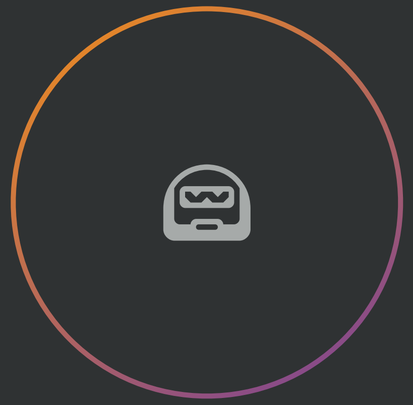
# CoffeeScript



hubot.github.com



github.com/hubot-scripts



# Super easy to set up

# Deploy to Heroku

README.md

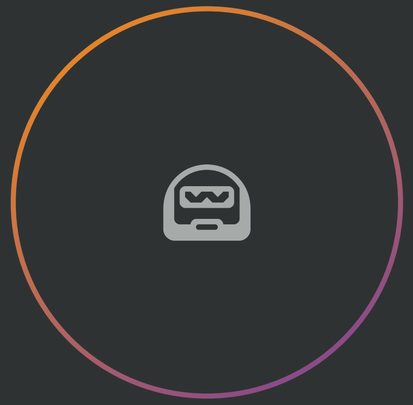
## Hubot, on Heroku, with the Slack adaptor

This is a version of GitHub's Campfire bot, Hubot, which is designed to be deployed on Heroku with the Slack adaptor. To deploy your own Hubot, just click the button below:

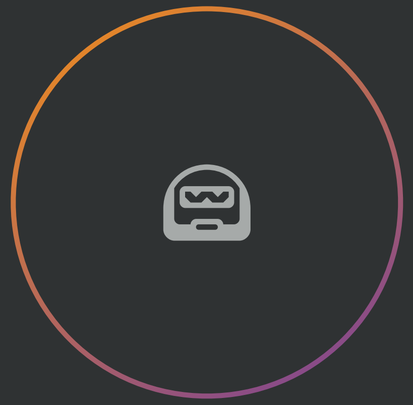
Deploy to Heroku

## License

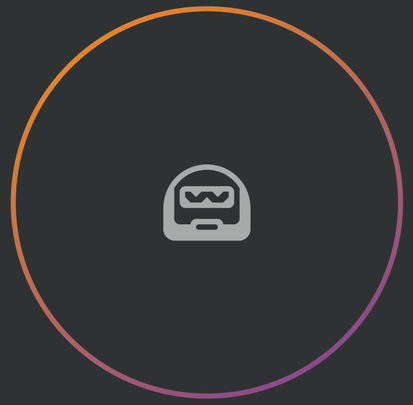
MIT, just like the upstream Hubot license, so go 🍻s.



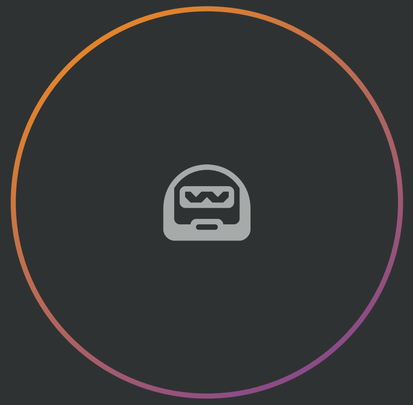
github.com/cobyism/hubot-heroku-slack



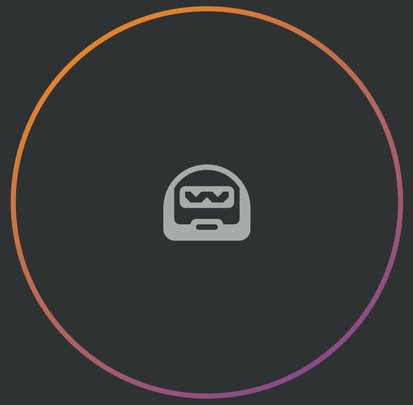
# What can hubot do?



850+ commands

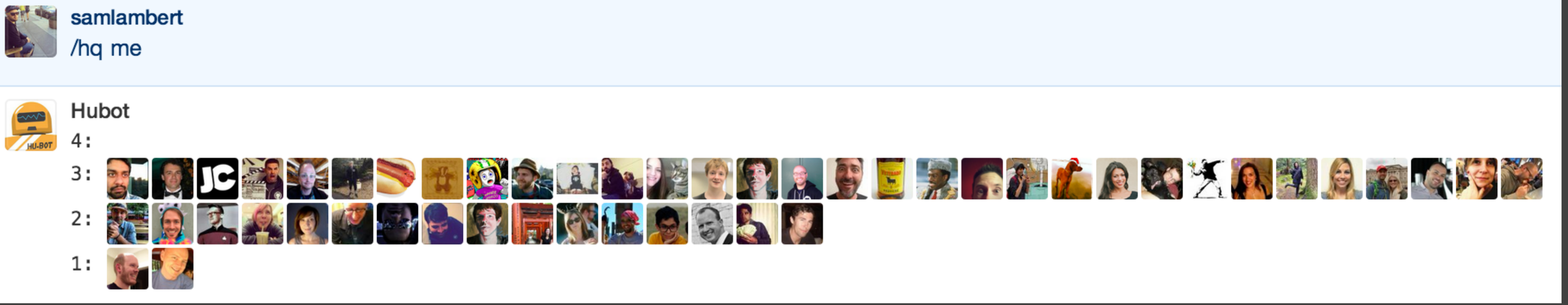


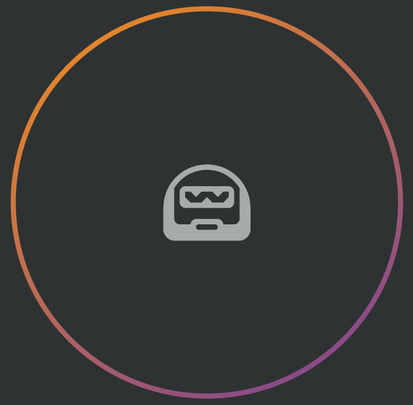
# Nearly everything



# Part of shipping is adding ChatOps

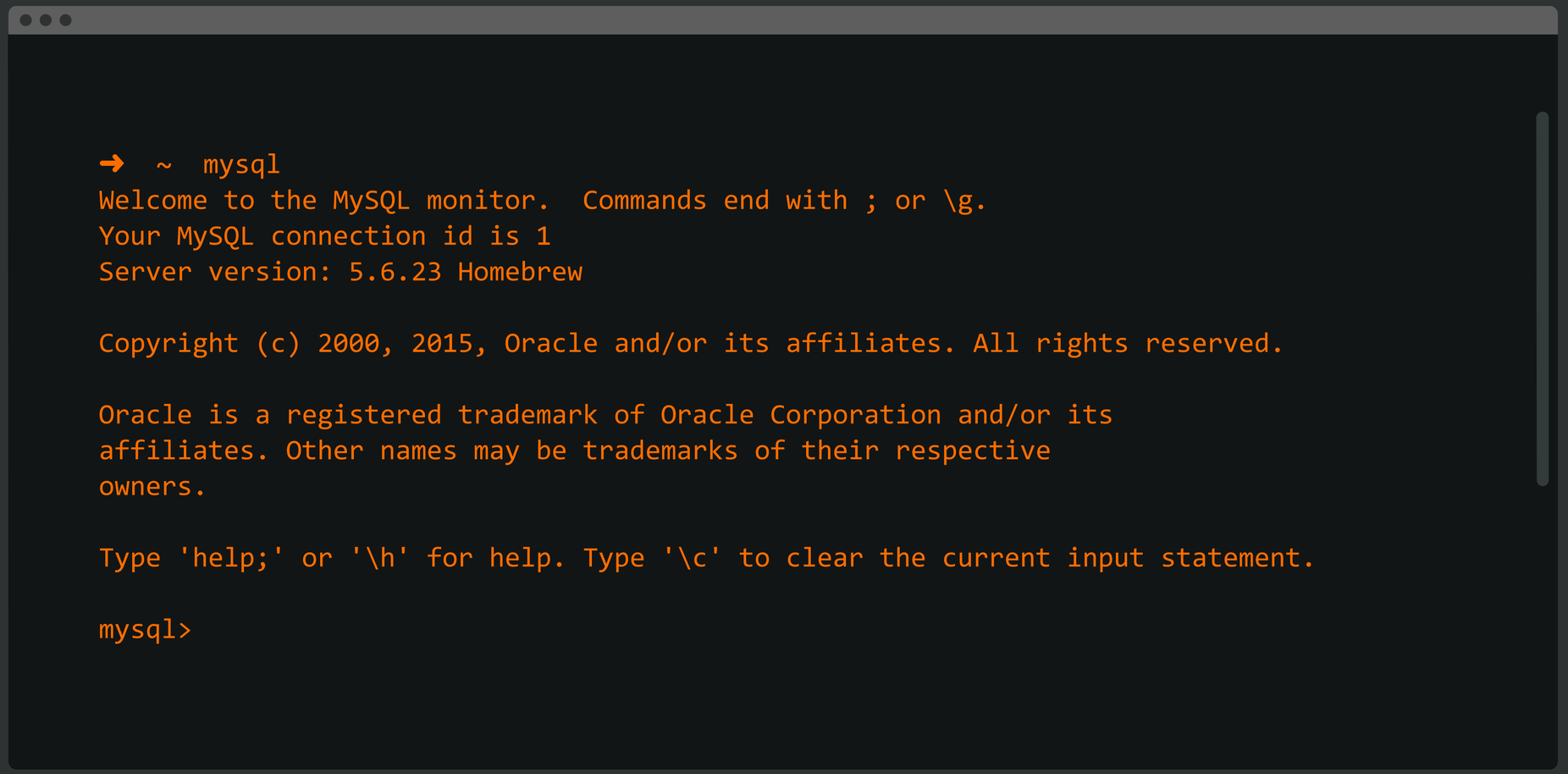
# /hq me



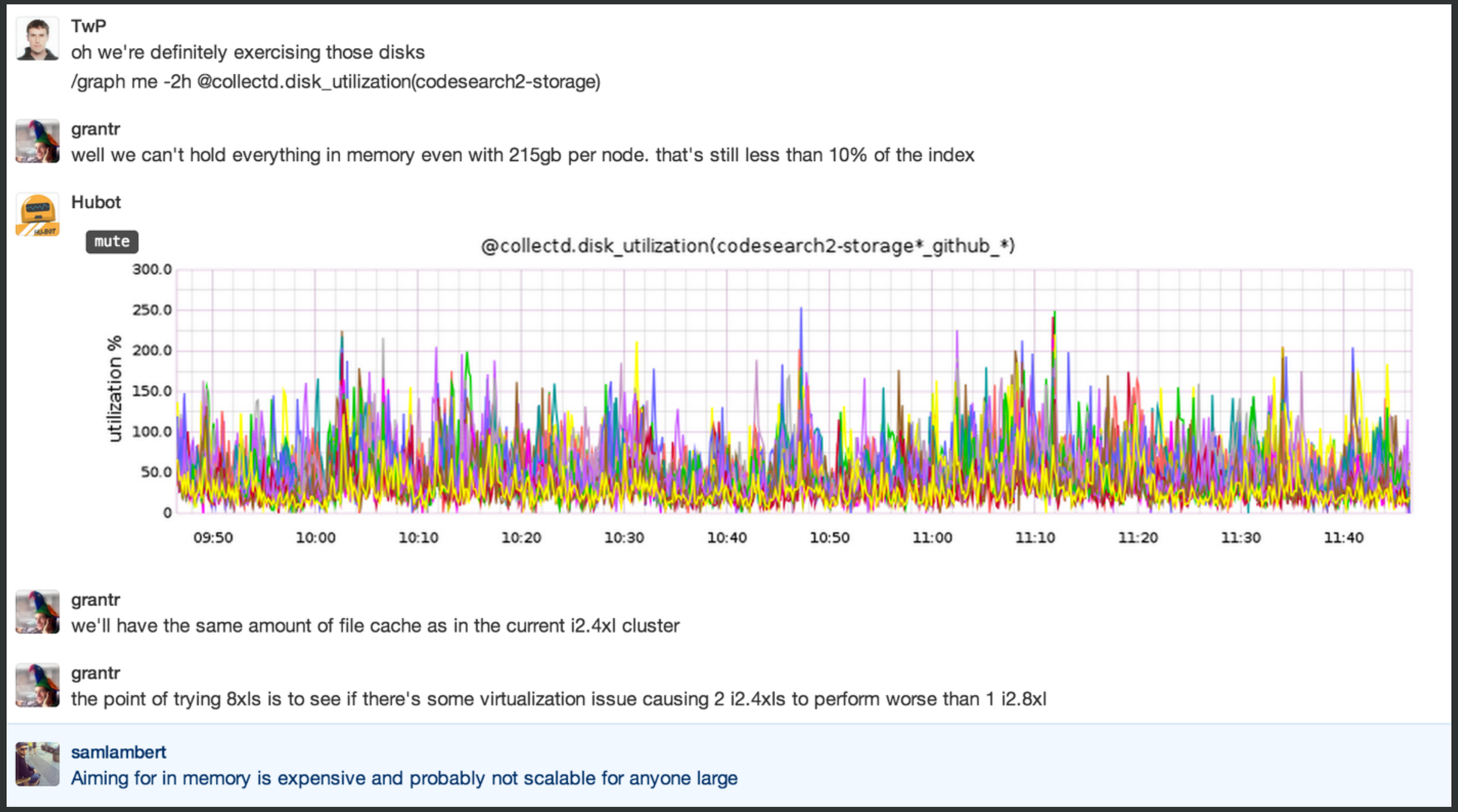


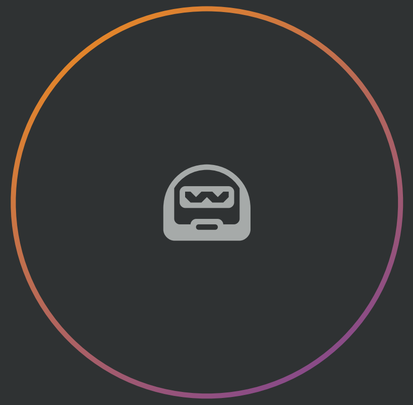
# ChatOps is core to our culture

# You are on your own

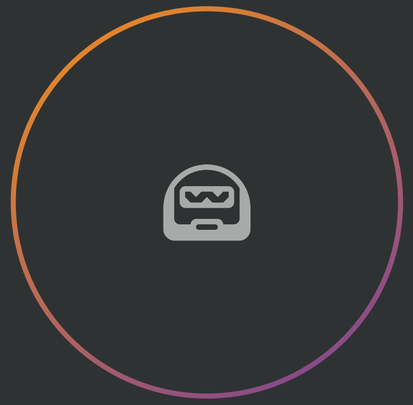


# Invite your friends

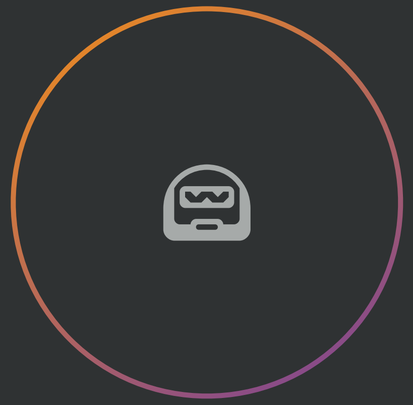




# It's all about context

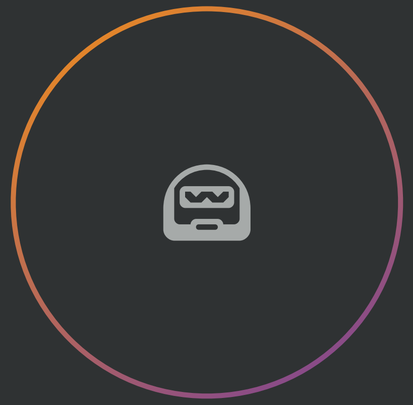


# Context is implicit

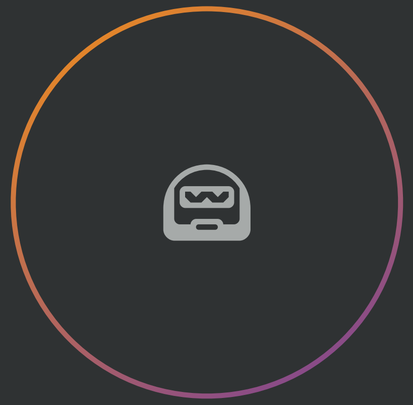


# Incident responce

# DDoS

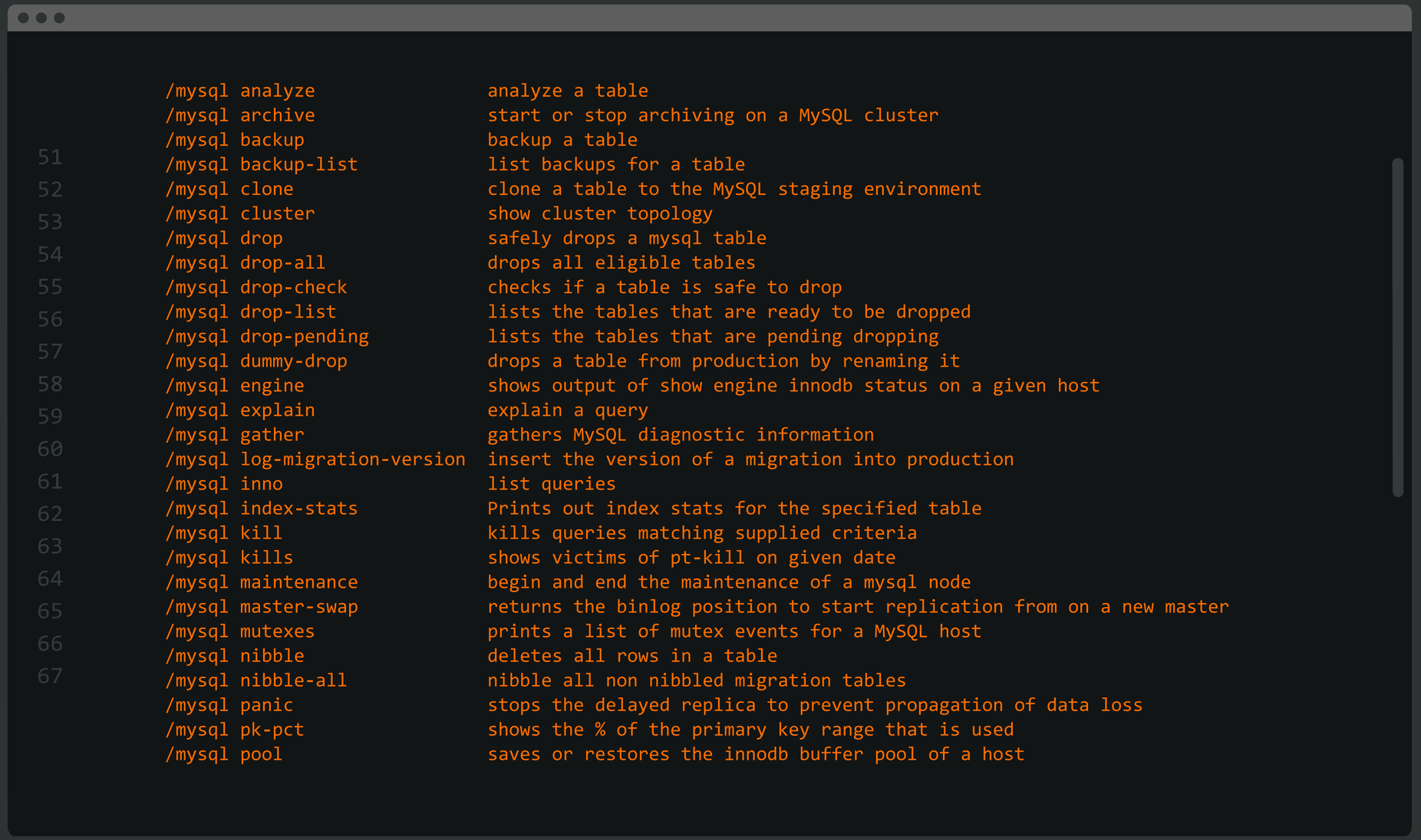


# /mitigation enable



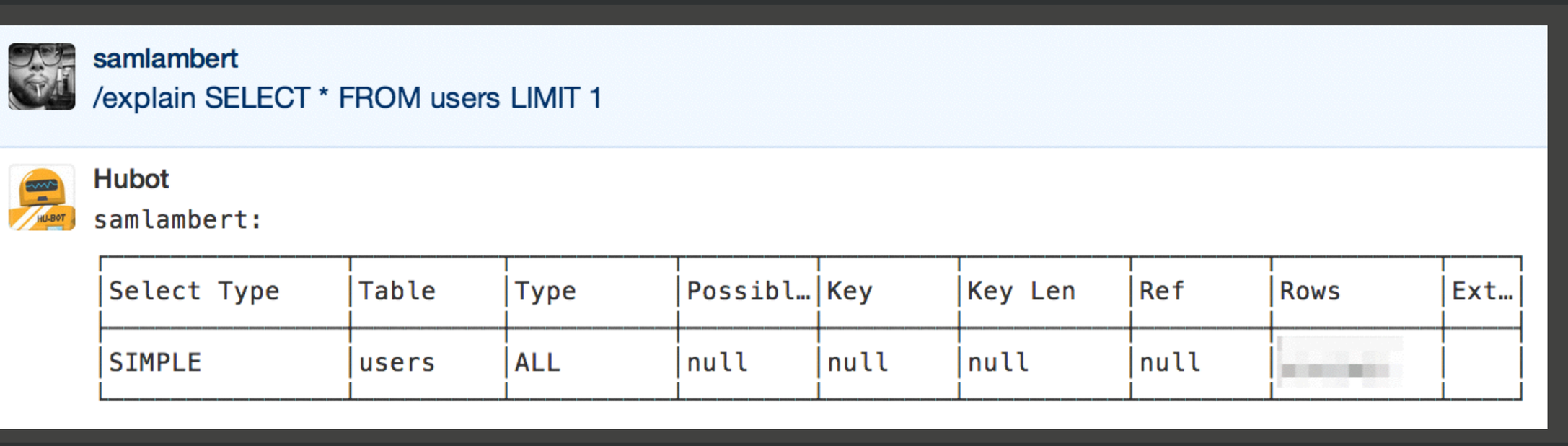
/mysql

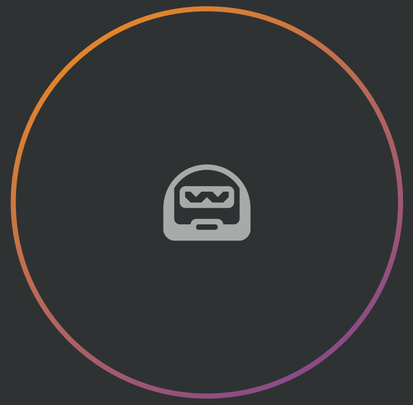
# /mysql



```
/mysql analyze                analyze a table
/mysql archive                start or stop archiving on a MySQL cluster
/mysql backup                 backup a table
/mysql backup-list            list backups for a table
/mysql clone                  clone a table to the MySQL staging environment
/mysql cluster                show cluster topology
/mysql drop                   safely drops a mysql table
/mysql drop-all               drops all eligible tables
/mysql drop-check             checks if a table is safe to drop
/mysql drop-list              lists the tables that are ready to be dropped
/mysql drop-pending           lists the tables that are pending dropping
/mysql dummy-drop             drops a table from production by renaming it
/mysql engine                 shows output of show engine innodb status on a given host
/mysql explain                explain a query
/mysql gather                 gathers MySQL diagnostic information
/mysql log-migration-version  insert the version of a migration into production
/mysql inno                   list queries
/mysql index-stats            Prints out index stats for the specified table
/mysql kill                   kills queries matching supplied criteria
/mysql kills                  shows victims of pt-kill on given date
/mysql maintenance            begin and end the maintenance of a mysql node
/mysql master-swap            returns the binlog position to start replication from on a new master
/mysql mutexes                prints a list of mutex events for a MySQL host
/mysql nibble                 deletes all rows in a table
/mysql nibble-all             nibble all non nibbled migration tables
/mysql panic                  stops the delayed replica to prevent propagation of data loss
/mysql pk-pct                 shows the % of the primary key range that is used
/mysql pool                   saves or restores the innodb buffer pool of a host
```

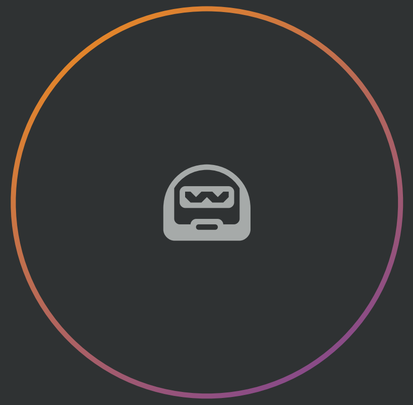
# /mysql explain





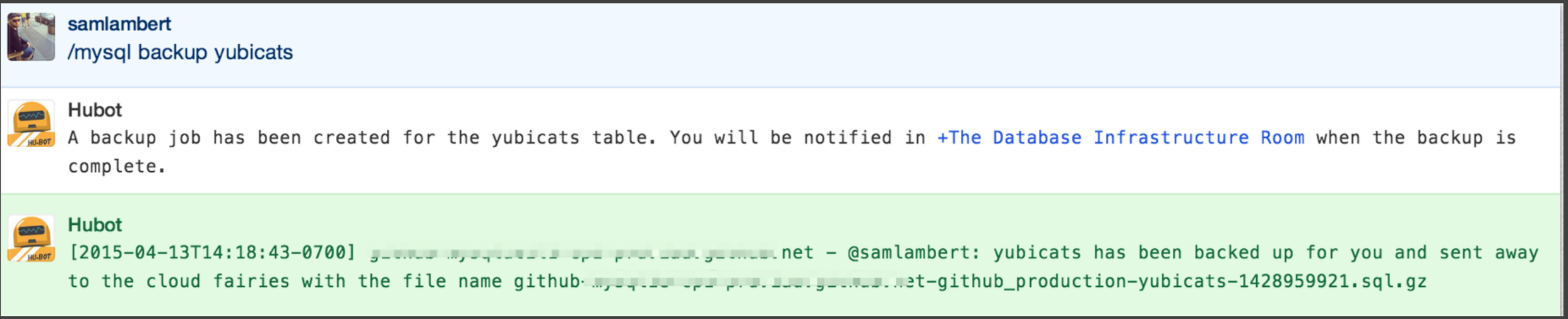
# Learn together

github.com/samlambert/hubot-mysql-chatops

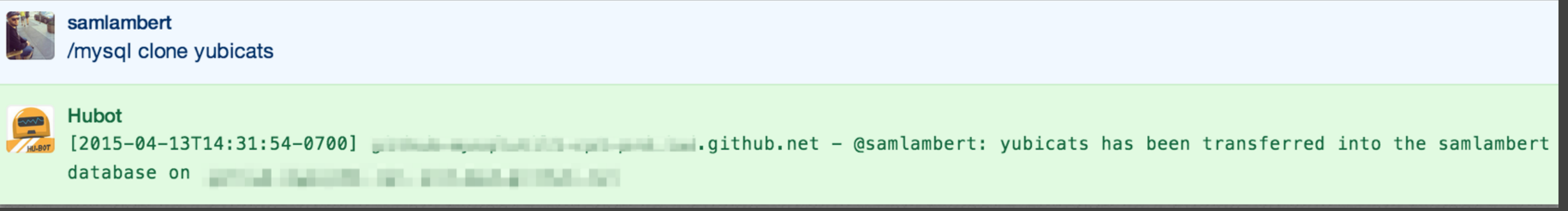


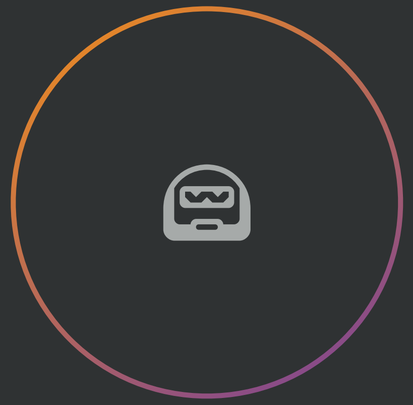
# Collaborative workspace

# /mysql backup



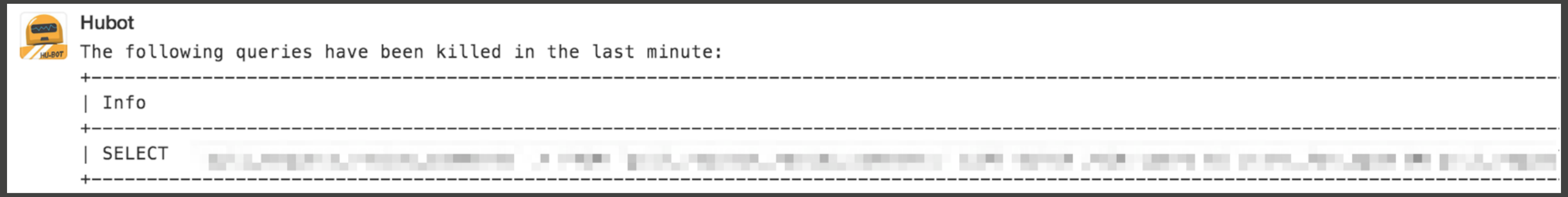
# /mysql clone

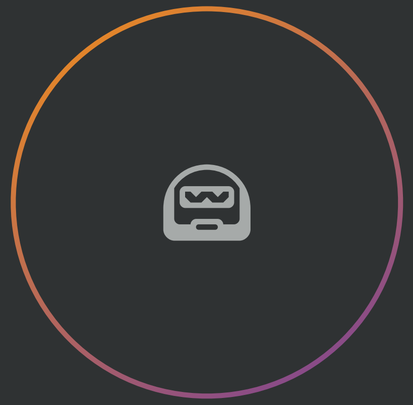

samlambert
/mysql clone yubicats
Hubot
[2015-04-13T14:31:54-0700] .github.net - @samlambert: yubicats has been transferred into the samlambert database on



# Killed queries

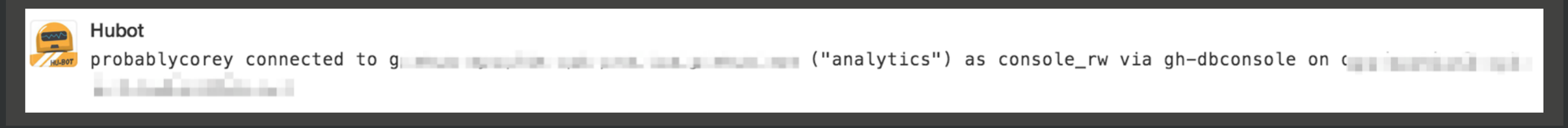
# Killed queries





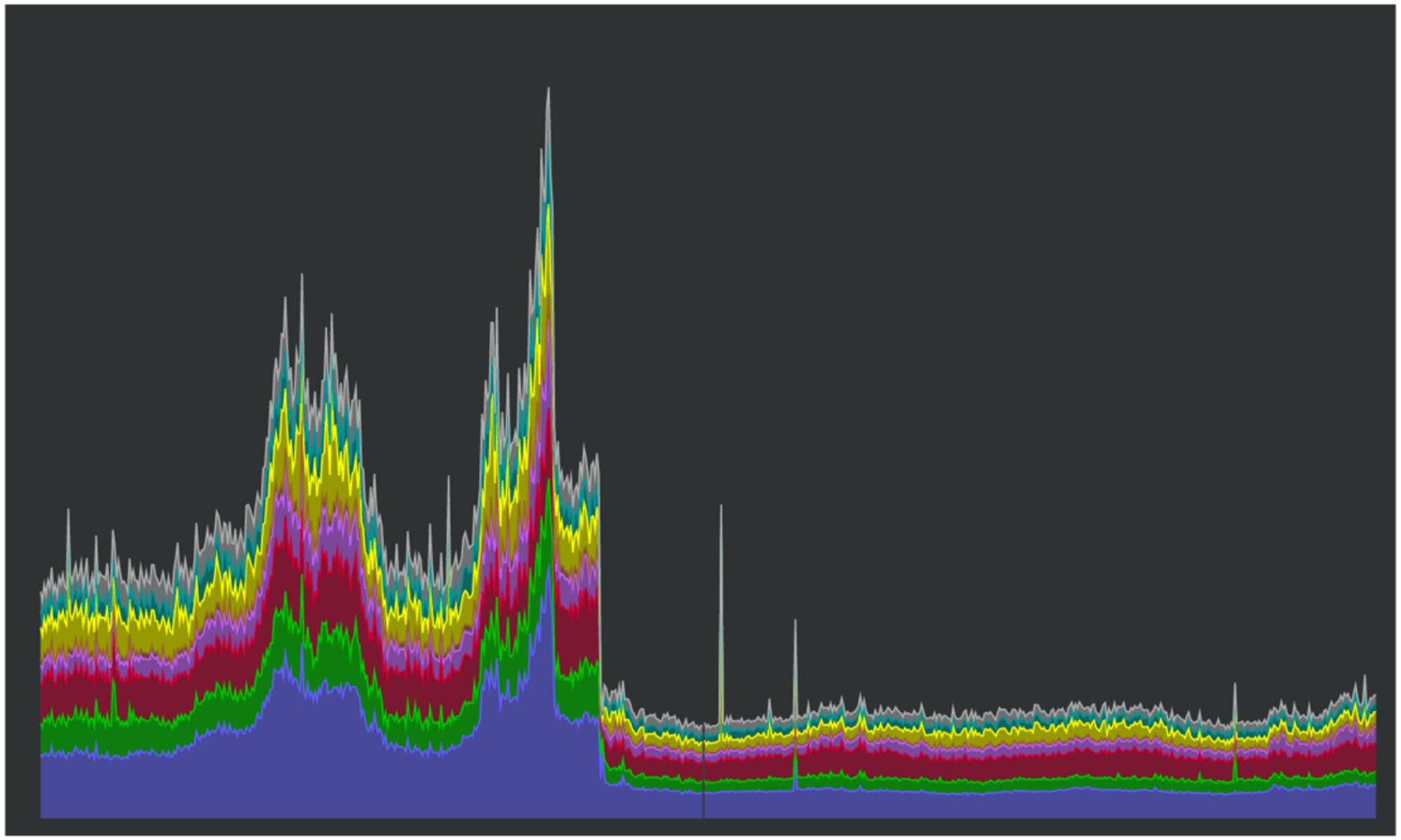
# User access

# User access


Hubot
probablycorey connected to g ("analytics") as console_rw via gh-dbconsole on c

# Graphme

# Graphme



Graphite Editor

Use shift-tab for autocomplete | General API | Target API

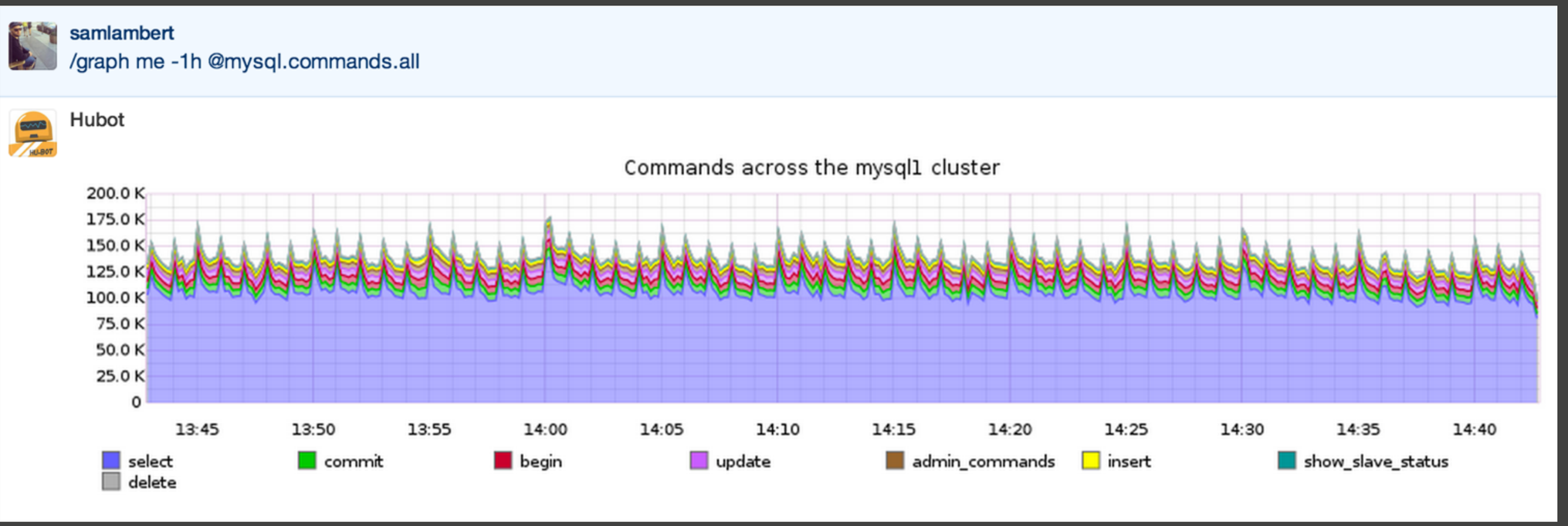
```
from: "-570d"
until: "-575d"
max: 200
areaMode: "stacked"
areaAlpha: "0.5"
hideGrid: true
bgcolor: "2F3233"
thickness: 1
fontSize: 15
hideLegend: "true"
hideAxes: true

template: "plain"
targets:
  "github.unicorn.{browser,anon,api,raw,googleb
    summarize: ["1min","avg"]
    substr: [2,3]
  "gitauth.unicorn.gitauth.mysql.time.mean":
    summarize: ["1min","avg"]
    substr: [2,3]
```

Delete Save

Hubot

```
/graph me -570d..-575d substr(summarize(github.u
```

# /graph me


samlambert
/graph me -1h @mysql.commands.all
Hubot
Commands across the mysql1 cluster
200.0 K
175.0 K
150.0 K
125.0 K
100.0 K
75.0 K
50.0 K
25.0 K
0
13:45
13:50
13:55
14:00
14:05
14:10
14:15
14:20
14:25
14:30
14:35
14:40
select
commit
begin
update
admin_commands
insert
show_slave_status
delete

# Amen

# Amen

- Monitoring app
- Set up backends
- Easily gather metrics
- Pushed to Graphme

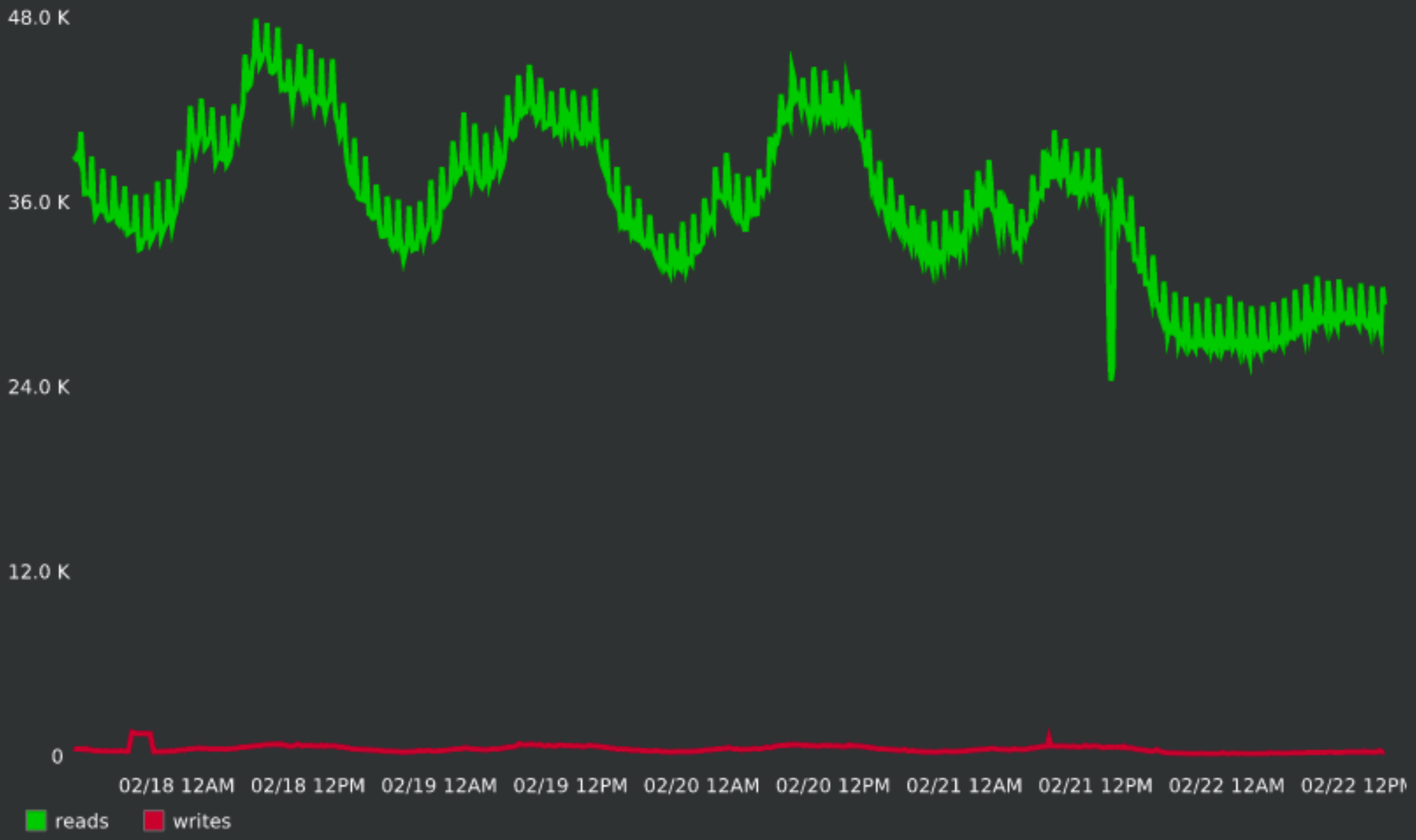
# We have these tools. Let's use them

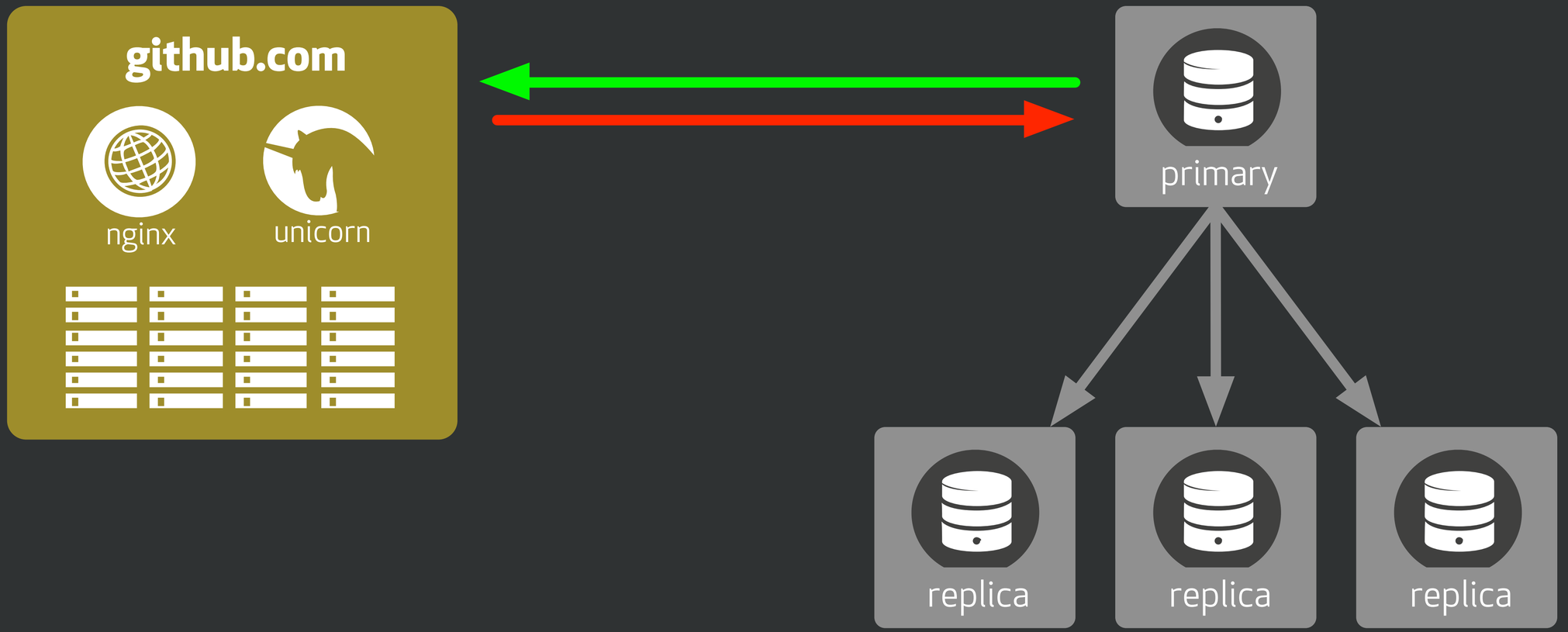
# It's about the app

# Let's talk strategy

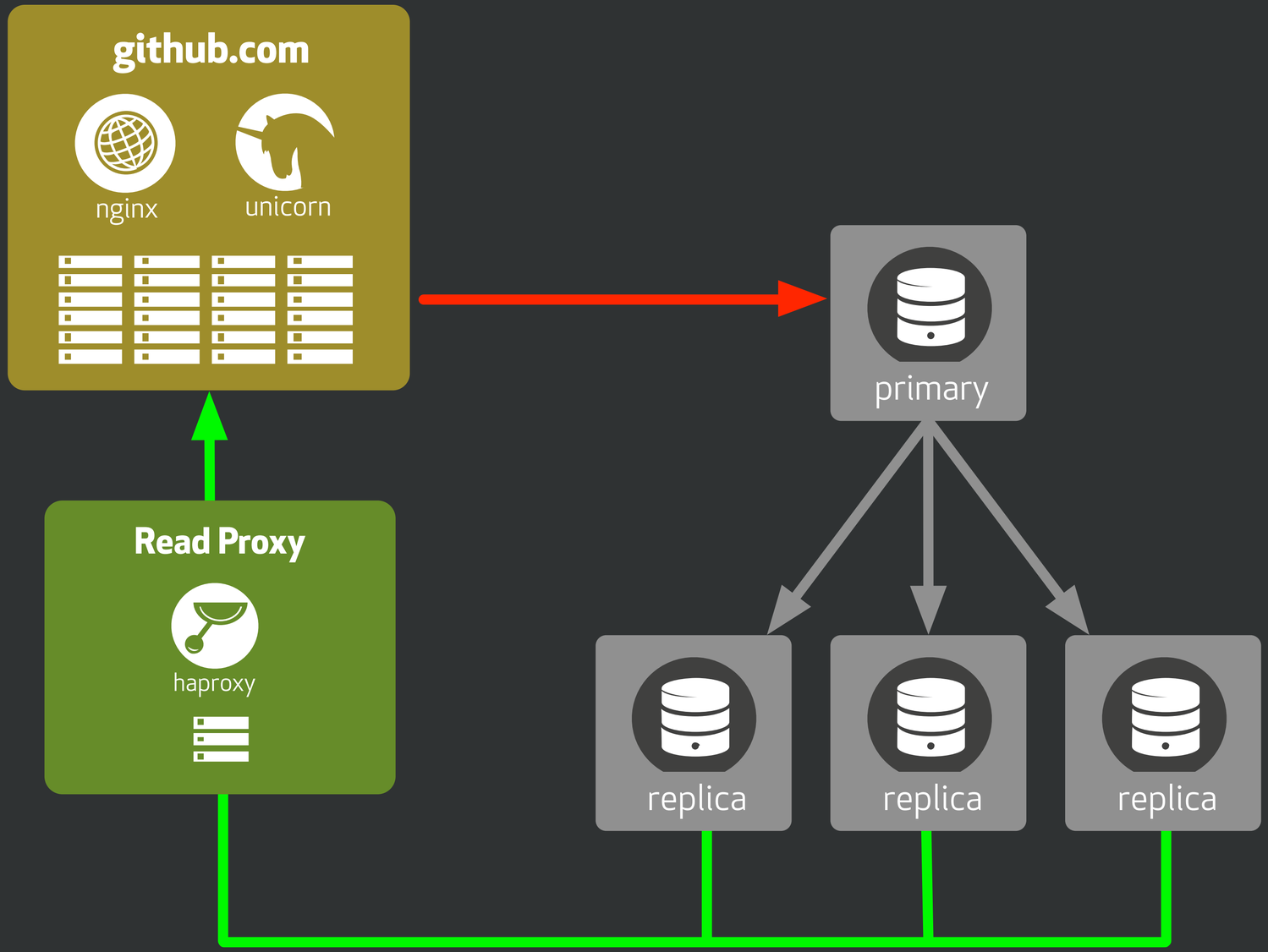
# All the hardware

# Single host


48.0 K
36.0 K
24.0 K
12.0 K
0
02/18 12AM
02/18 12PM
02/19 12AM
02/19 12PM
02/20 12AM
02/20 12PM
02/21 12AM
02/21 12PM
02/22 12AM
02/22 12PM
reads
writes


github.com
nginx
unicorn
primary
replica
replica
replica

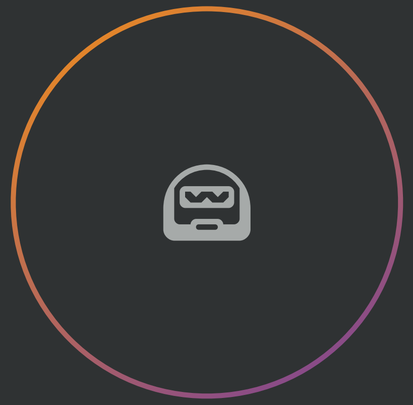
# Doesn't scale


github.com
nginx
unicorn
primary
Read Proxy
haproxy
replica
replica
replica

# Use existing patterns

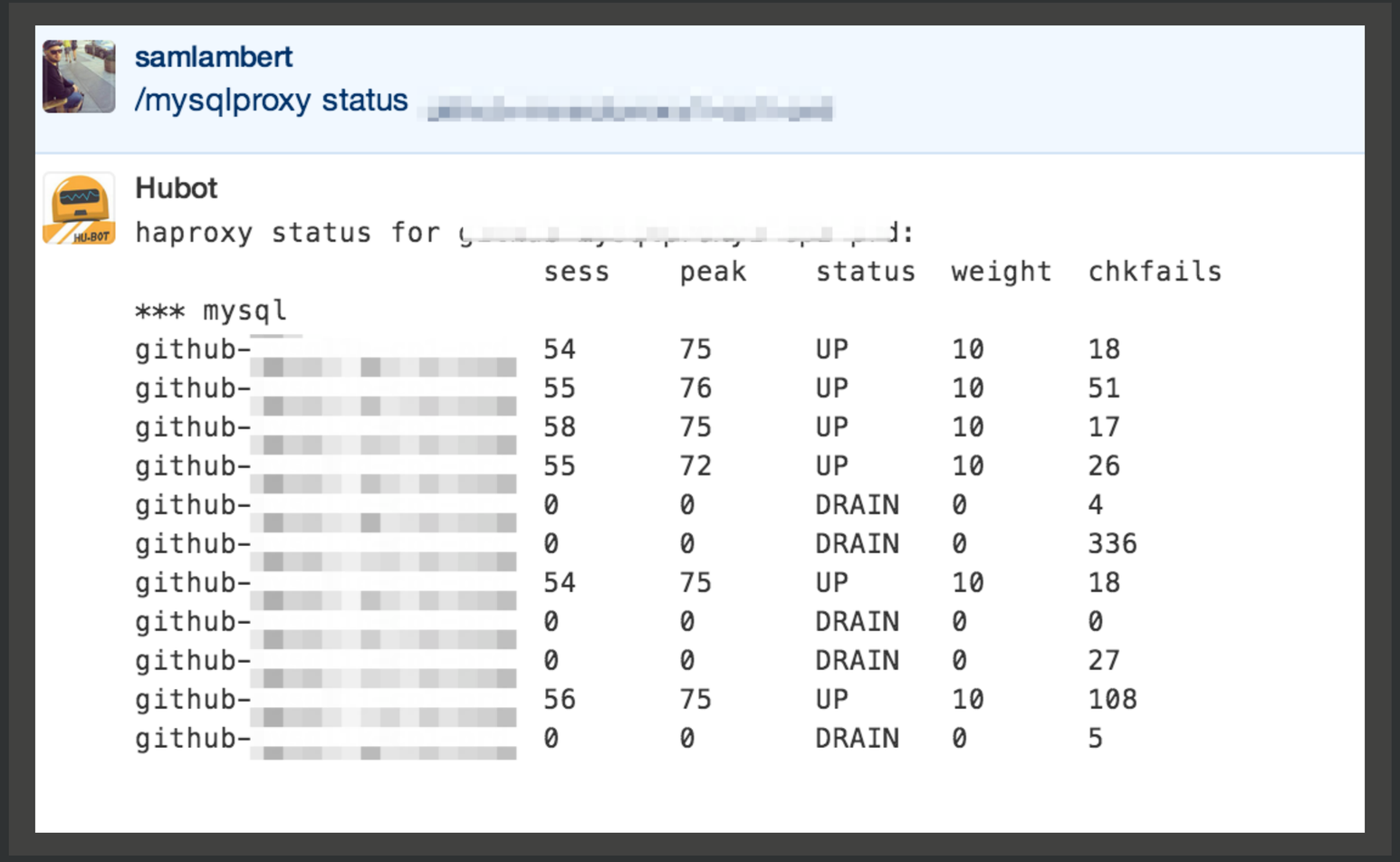
# Tuning Haproxy

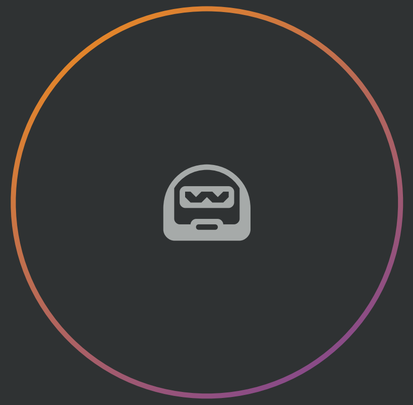
# App failover logic



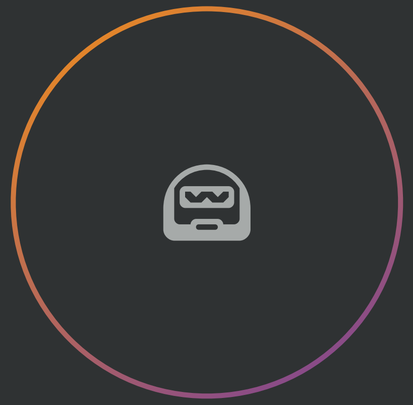
/mysqlproxy

# /mysqlproxy status

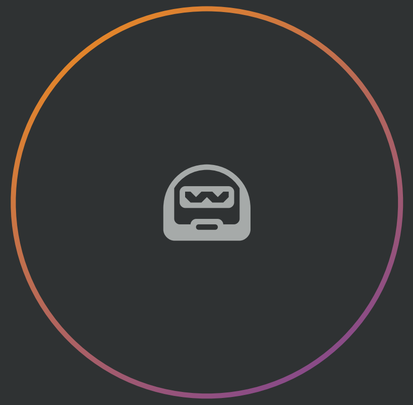




# /mysqlproxy weight



/mysqlproxy disable



# /mysqlproxy enable

# App refactor

# Send all GET requests to a replica

# Stick to master for a second after POST

# How do you refactor these changes?

# Some GETs make writes

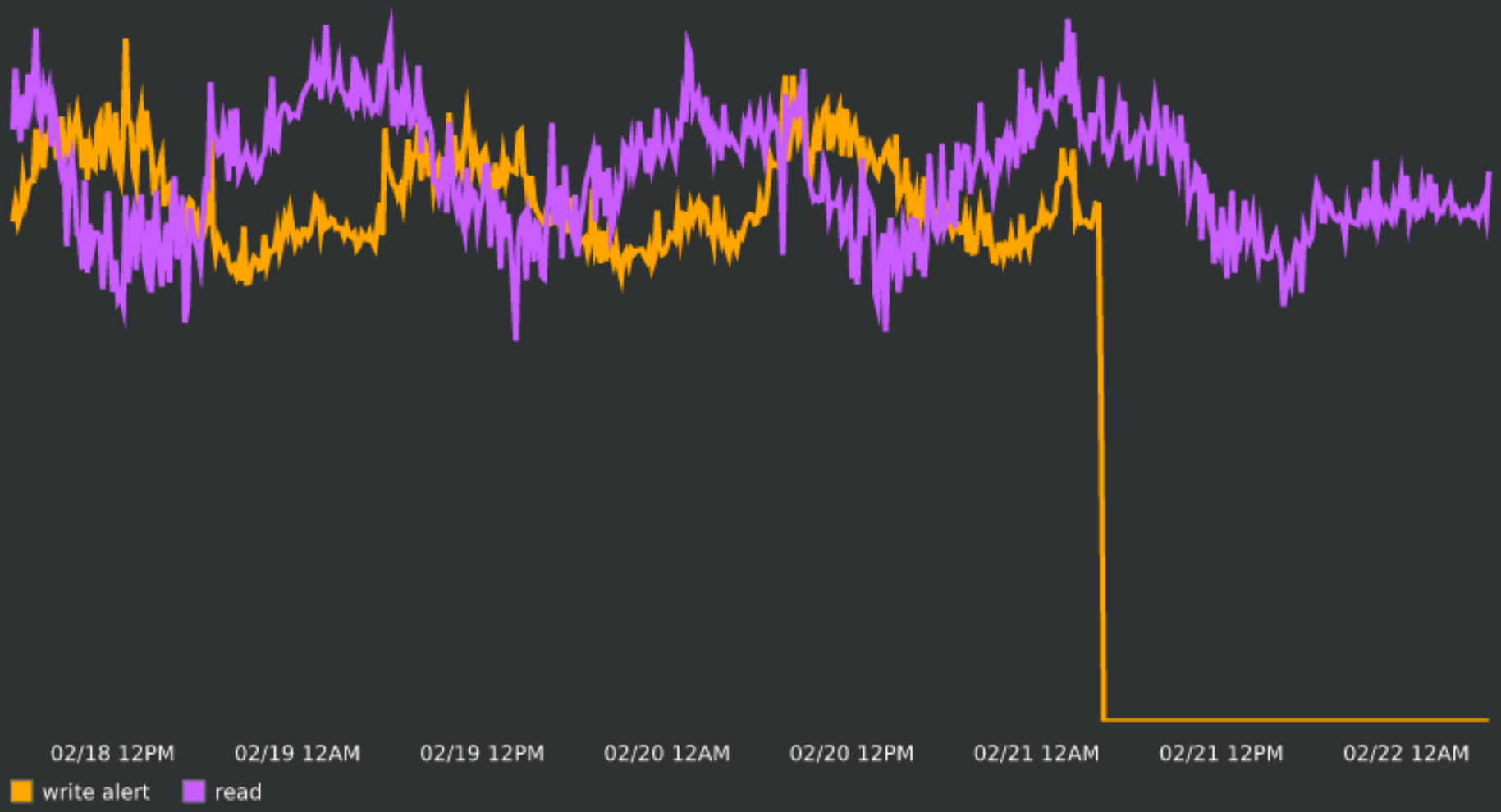
# Set up write alert connection

# Write alerts go to Haystack

# Get the stack trace

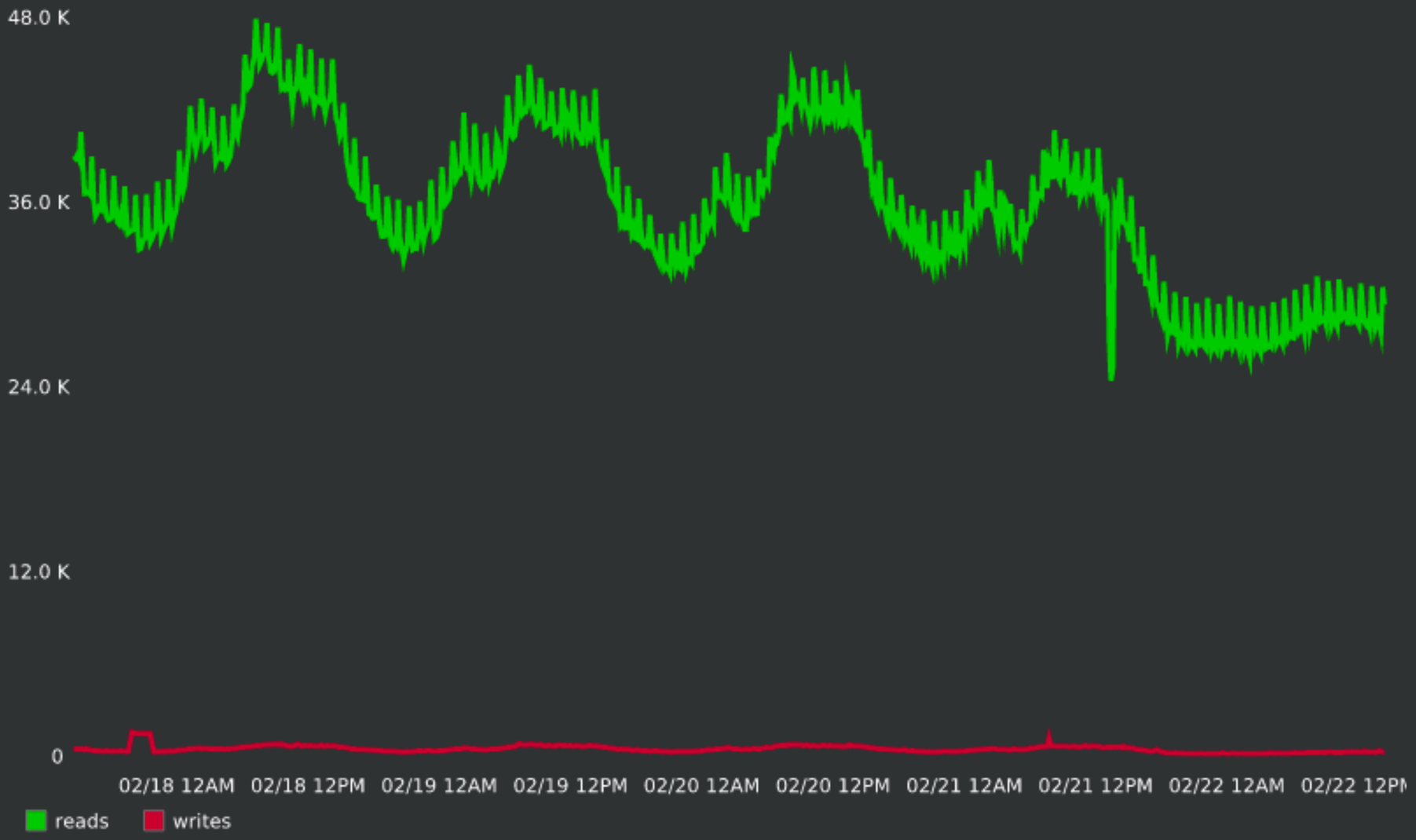
# Gradual iteration over the codebase

# Change

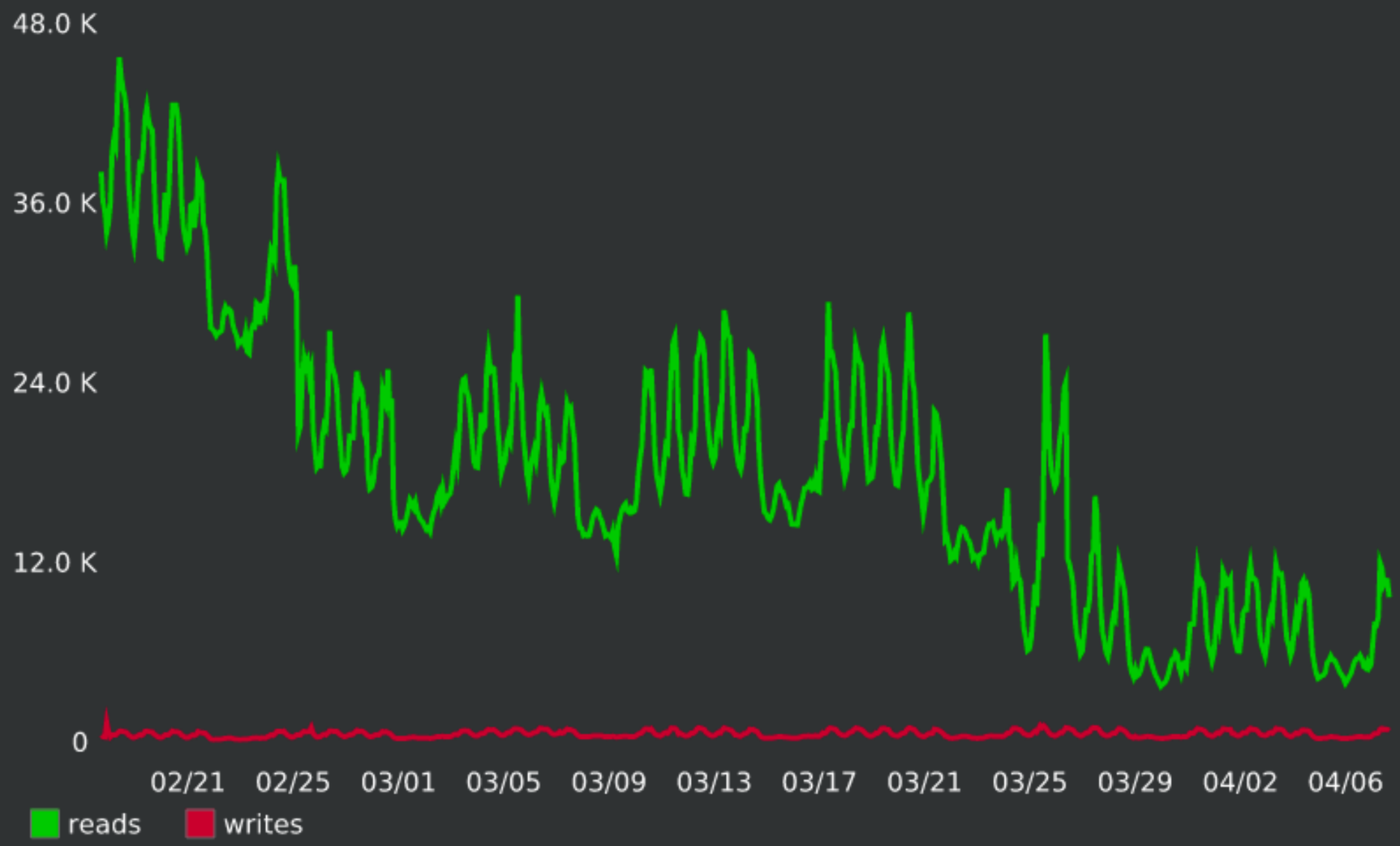

02/18 12PM
02/19 12AM
02/19 12PM
02/20 12AM
02/20 12PM
02/21 12AM
02/21 12PM
02/22 12AM
write alert
read

# Gitauth

# Single host

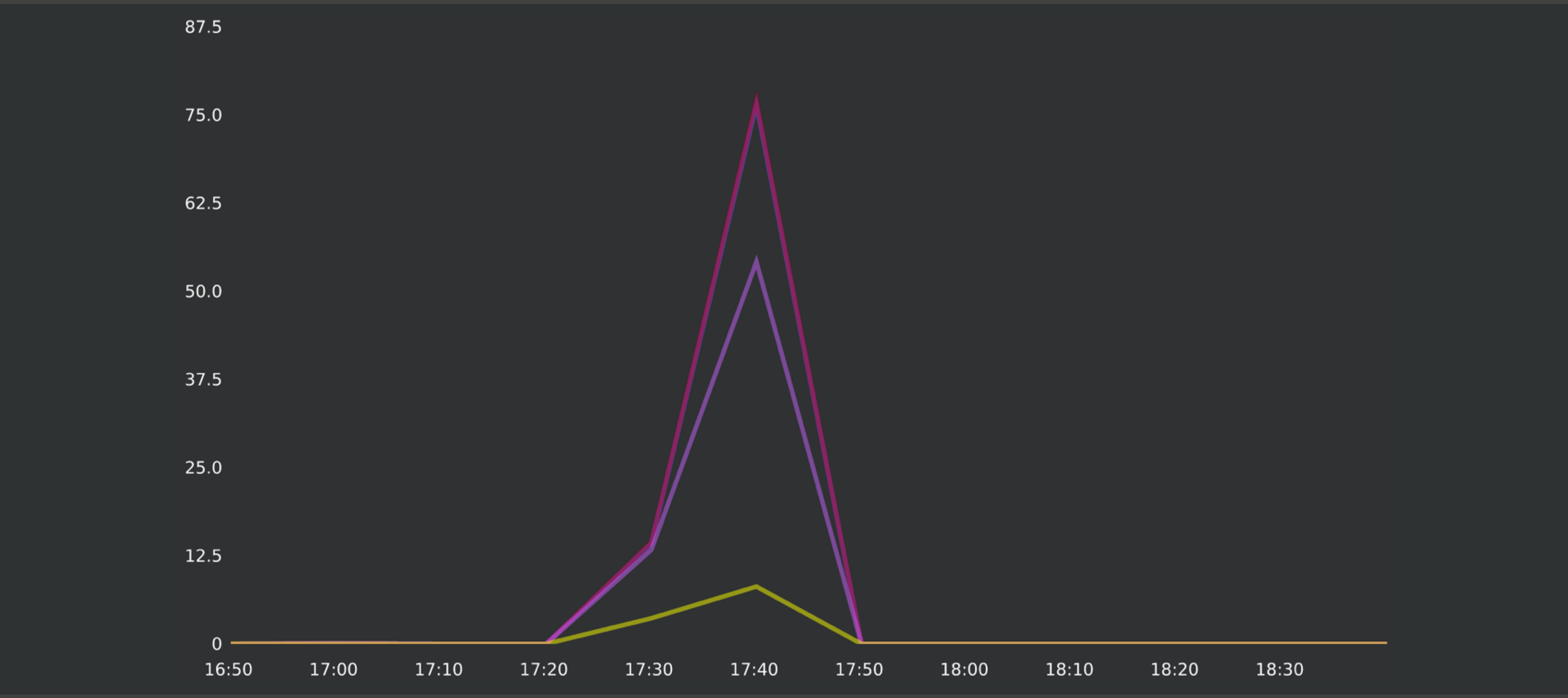

48.0 K
36.0 K
24.0 K
12.0 K
0
02/18 12AM
02/18 12PM
02/19 12AM
02/19 12PM
02/20 12AM
02/20 12PM
02/21 12AM
02/21 12PM
02/22 12AM
02/22 12PM
reads
writes

# Master


48.0 K
36.0 K
24.0 K
12.0 K
0
02/21
02/25
03/01
03/05
03/09
03/13
03/17
03/21
03/25
03/29
04/02
04/06
reads
writes

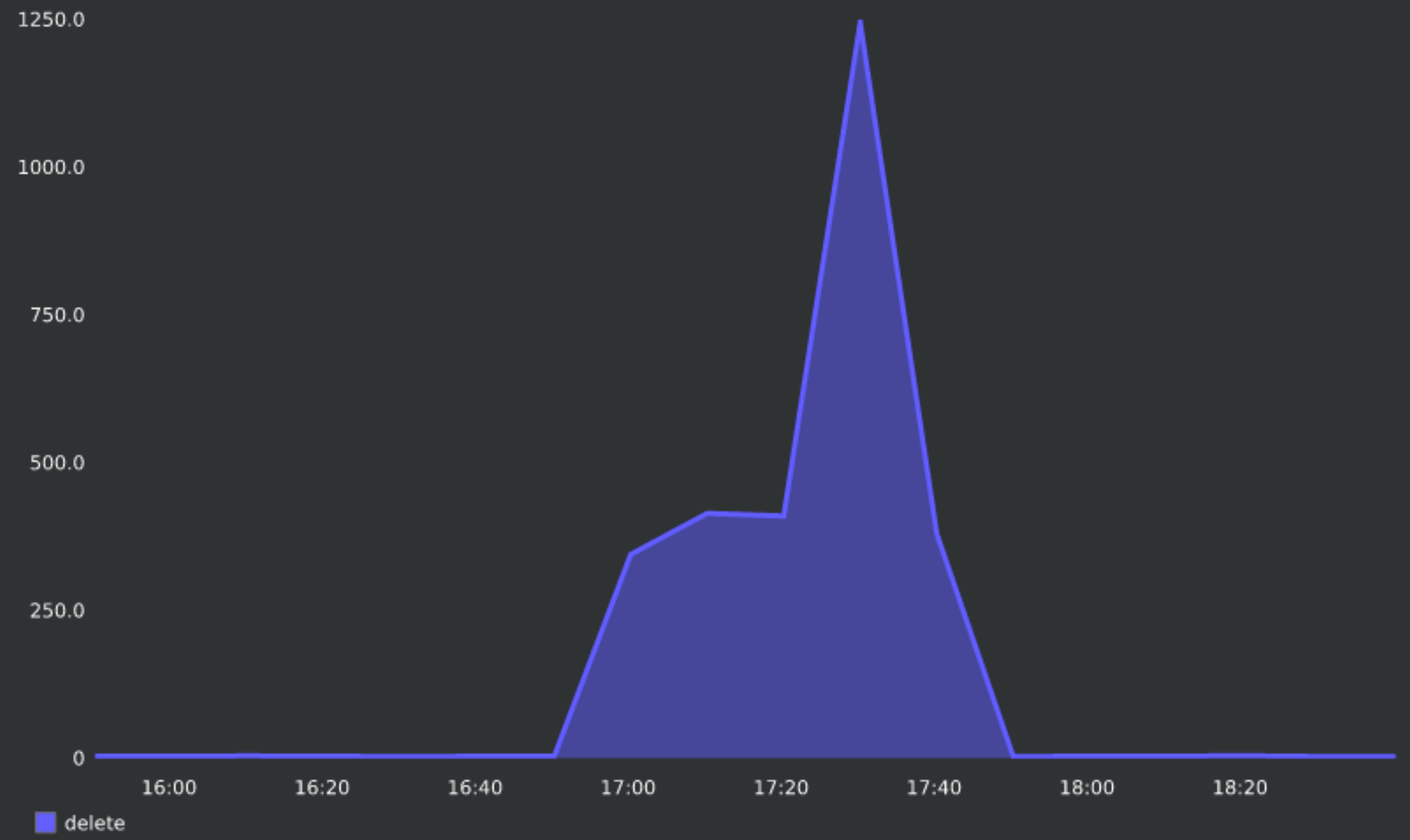
# Now we are sensitive to delay

# Delay!!


87.5
75.0
62.5
50.0
37.5
25.0
12.5
0
16:50
17:00
17:10
17:20
17:30
17:40
17:50
18:00
18:10
18:20
18:30

# What is happening?

# Delay!!

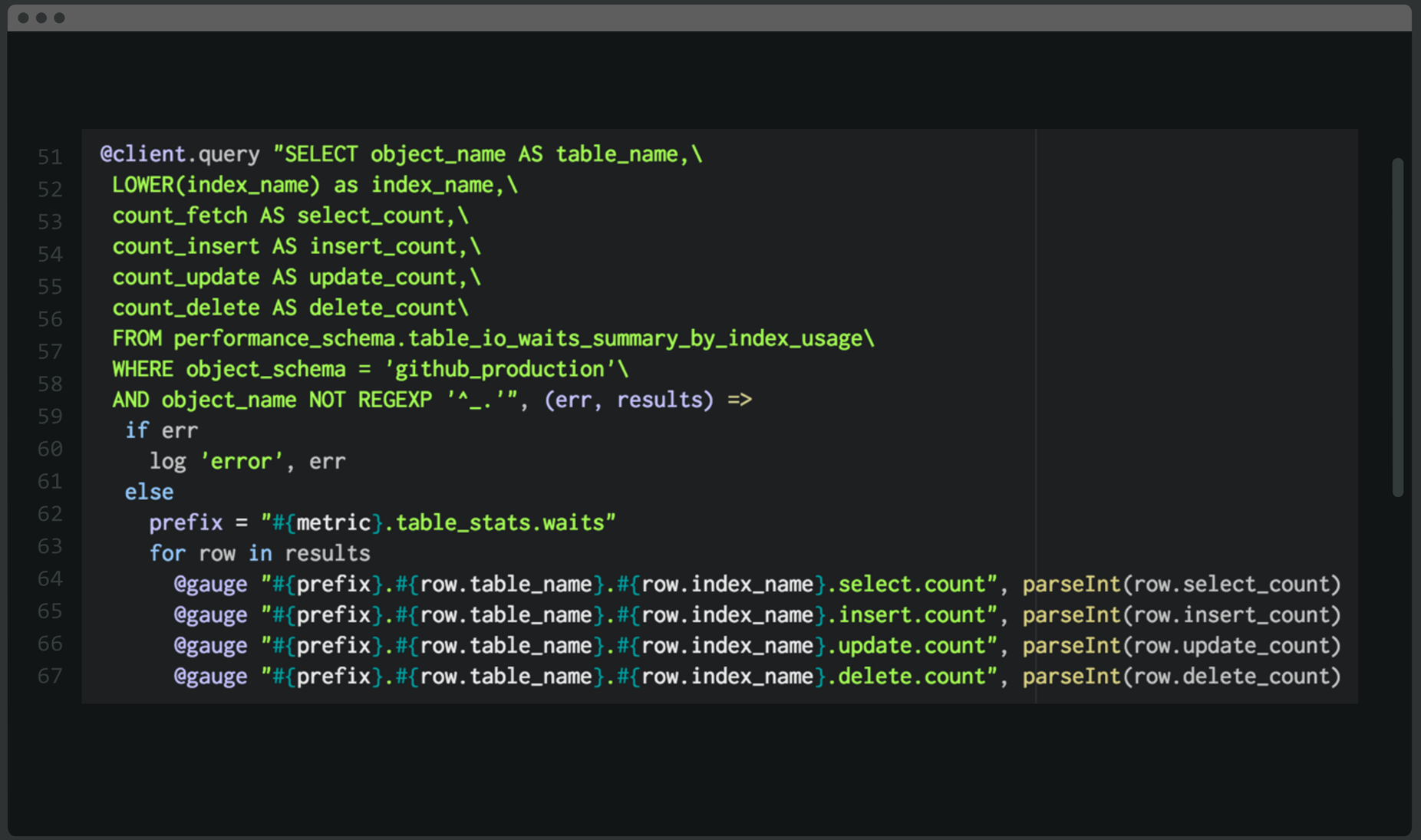

1250.0
1000.0
750.0
500.0
250.0
0
16:00
16:20
16:40
17:00
17:20
17:40
18:00
18:20
delete

# Ok, so deletes are happening

# Attack on two fronts

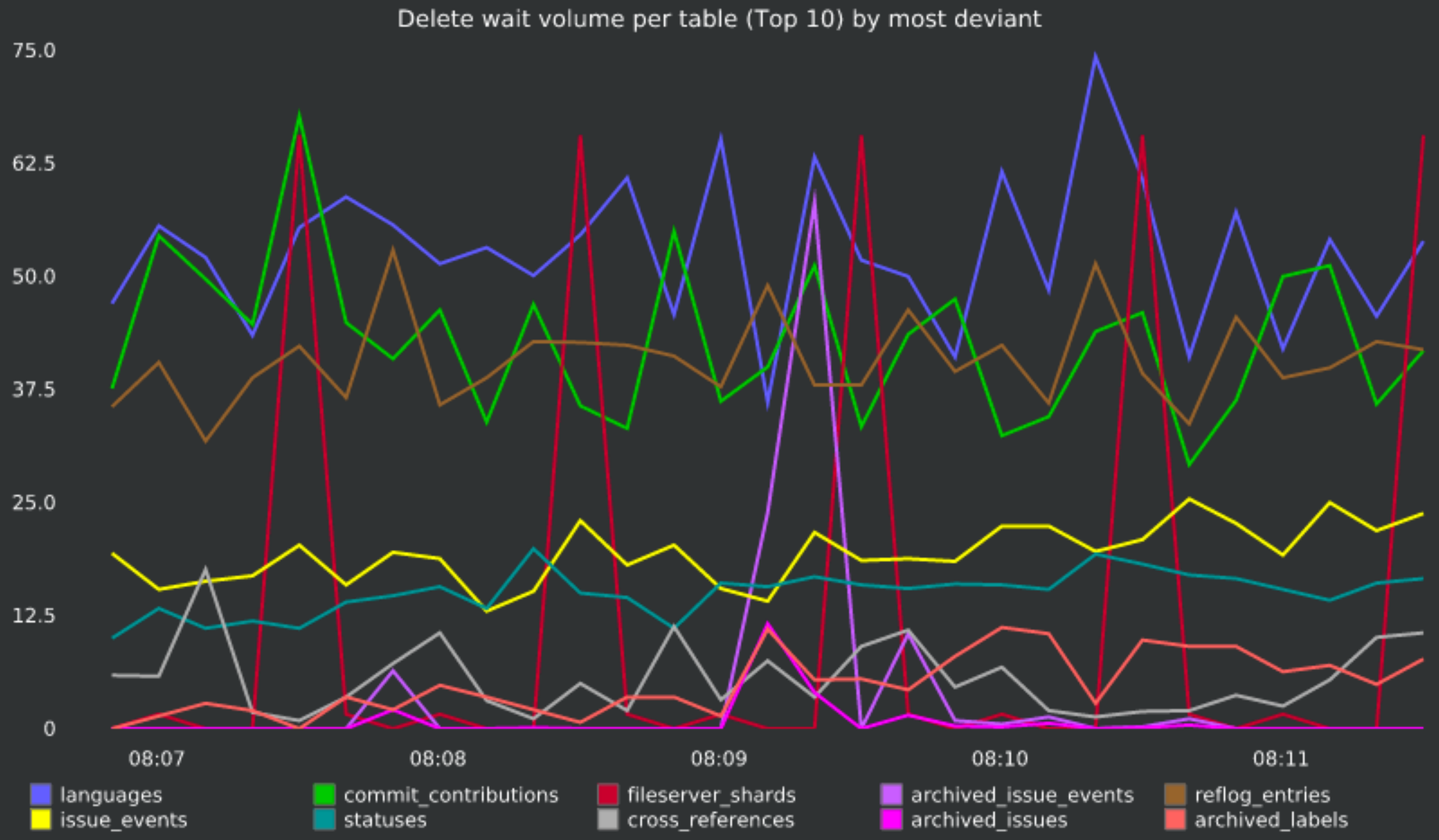
# Performance schema

# Amen - IO waits by index



```
@client.query "SELECT object_name AS table_name,\
 LOWER(index_name) as index_name,\
 count_fetch AS select_count,\
 count_insert AS insert_count,\
 count_update AS update_count,\
 count_delete AS delete_count\
 FROM performance_schema.table_io_waits_summary_by_index_usage\
 WHERE object_schema = 'github_production'\
 AND object_name NOT REGEXP '^_.'", (err, results) =>
  if err
    log 'error', err
  else
    prefix = "#{metric}.table_stats.waits"
    for row in results
      @gauge "#{prefix}.#{row.table_name}.#{row.index_name}.select.count", parseInt(row.select_count)
      @gauge "#{prefix}.#{row.table_name}.#{row.index_name}.insert.count", parseInt(row.insert_count)
      @gauge "#{prefix}.#{row.table_name}.#{row.index_name}.update.count", parseInt(row.update_count)
      @gauge "#{prefix}.#{row.table_name}.#{row.index_name}.delete.count", parseInt(row.delete_count)
```

# Graphme


Delete wait volume per table (Top 10) by most deviant
75.0
62.5
50.0
37.5
25.0
12.5
0
08:07
08:08
08:09
08:10
08:11
languages
issue_events
commit_contributions
statuses
fileserver_shards
cross_references
archived_issue_events
archived_issues
reflog_entries
archived_labels

# Haystack

# Affected rows bucket

# Github-query-affected_rows / QueryAffectedRowsLogger::QueryAffectedRows

October 16, 2014 08:35:22

Related needles

samlambert mentioned in Repository deletes create replication lag
October 16, 2014 08:47:39

github/github#34418

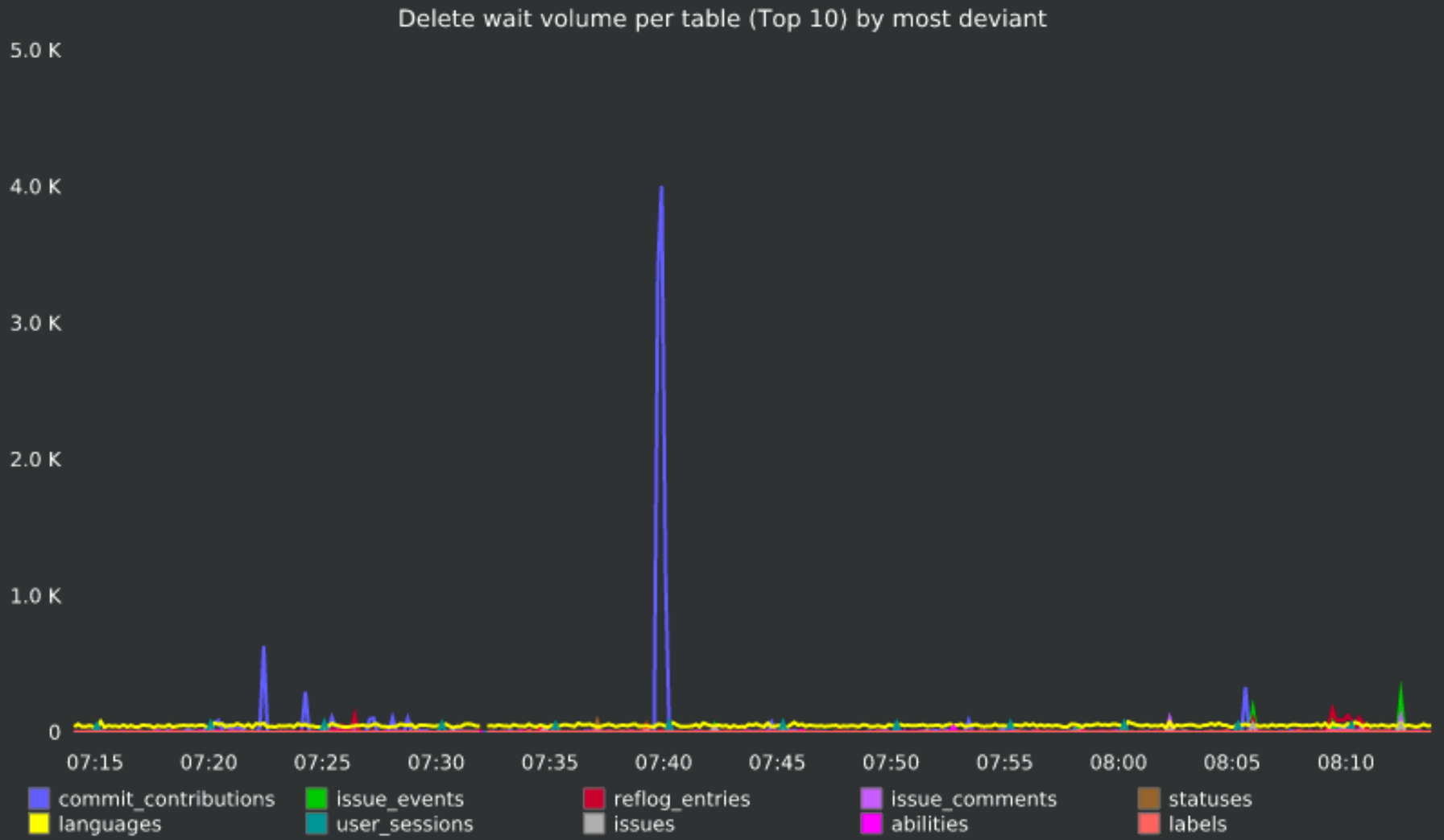
```
Changed 9665 rows
DELETE FROM `commit_contributions` WHERE ((repository_id = [redacted]))
```

| | |
|---|---|
| **class_name** | QueryAffectedRowsLogger::QueryAffectedRows |
| **current_ref** | master |
| **network_id** | [redacted] |
| **query** | DELETE FROM `commit_contributions` WHERE ((repository_id = [redacted])) |
| **queue_time** | 0.09425826991015625 |
| **queued_from** | Api::Repos#DELETE |
| **rails** | 3.0.20.github12 |
| **request_id** | [redacted] |
| **resque_args** | [25307443, nil, [redacted] |
| **resque_job** | GitHub::Jobs::RepositoryDelete |
| **ruby** | ruby 2.1.2p95-github-tcmalloc (71c08a81c0) [x86_64-linux] |
| **server** | [redacted] |

# Haystack

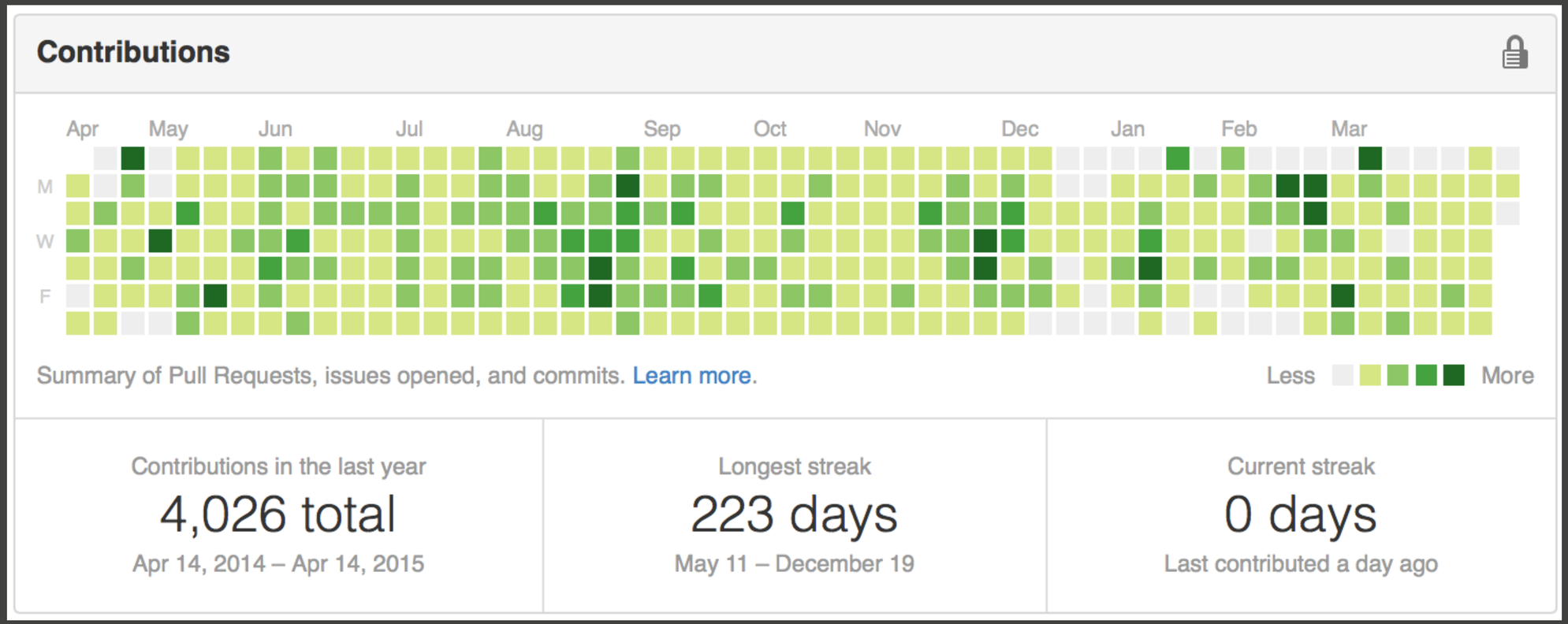
samlambert mentioned in Repository deletes create replication lag
October 16, 2014 08:47:39

# Graphme


Delete wait volume per table (Top 10) by most deviant
5.0 K
4.0 K
3.0 K
2.0 K
1.0 K
0
07:15
07:20
07:25
07:30
07:35
07:40
07:45
07:50
07:55
08:00
08:05
08:10
commit_contributions
issue_events
reflog_entries
issue_comments
statuses
languages
user_sessions
issues
abilities
labels

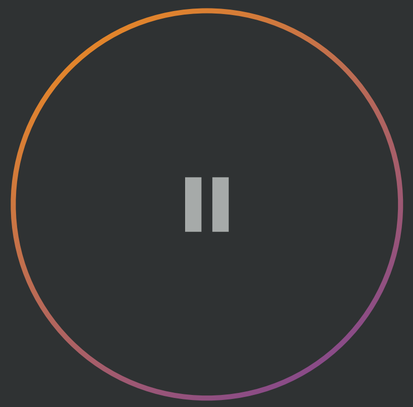
# Too many affected rows

# Contributions



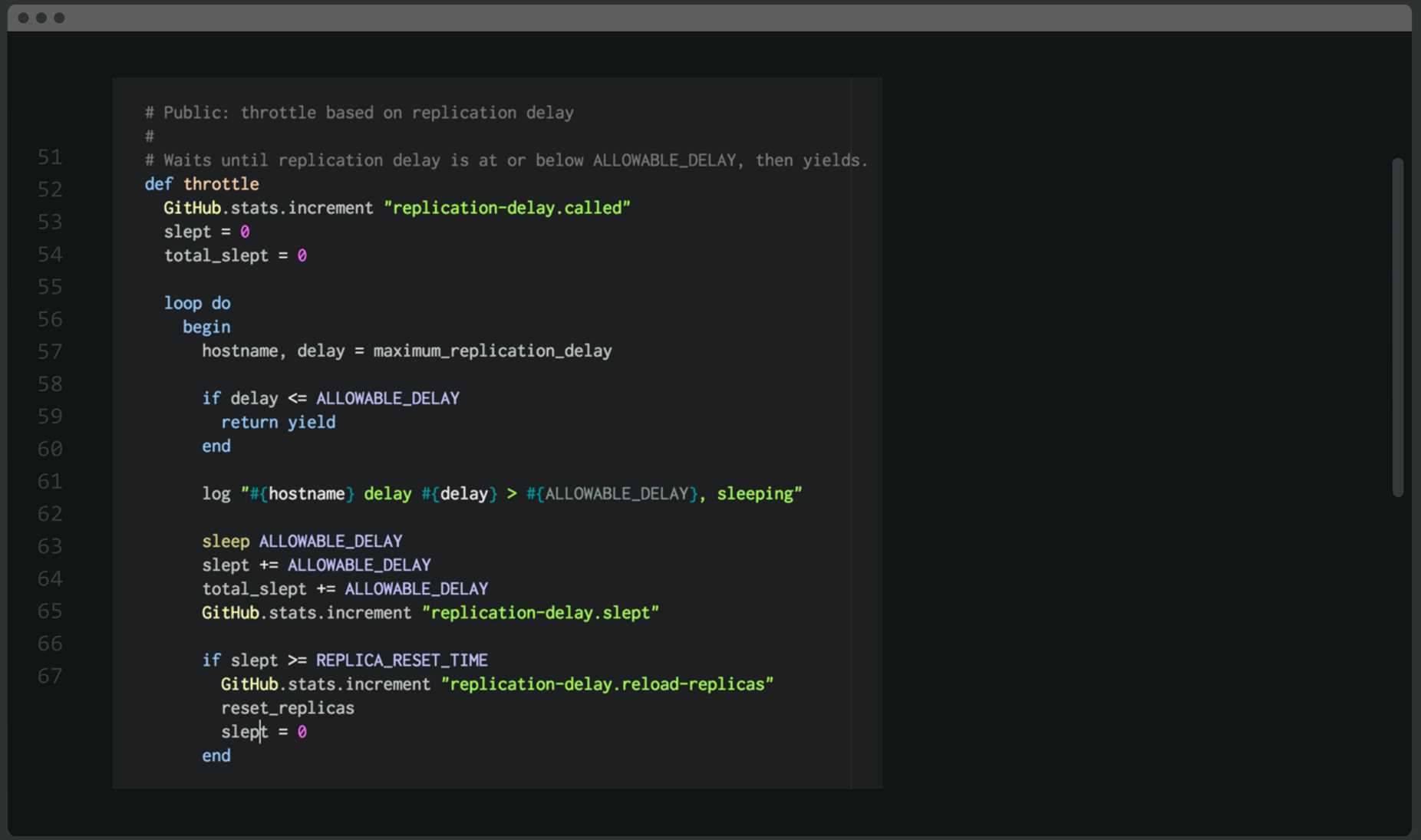
# Dependent destroy

# Problem found



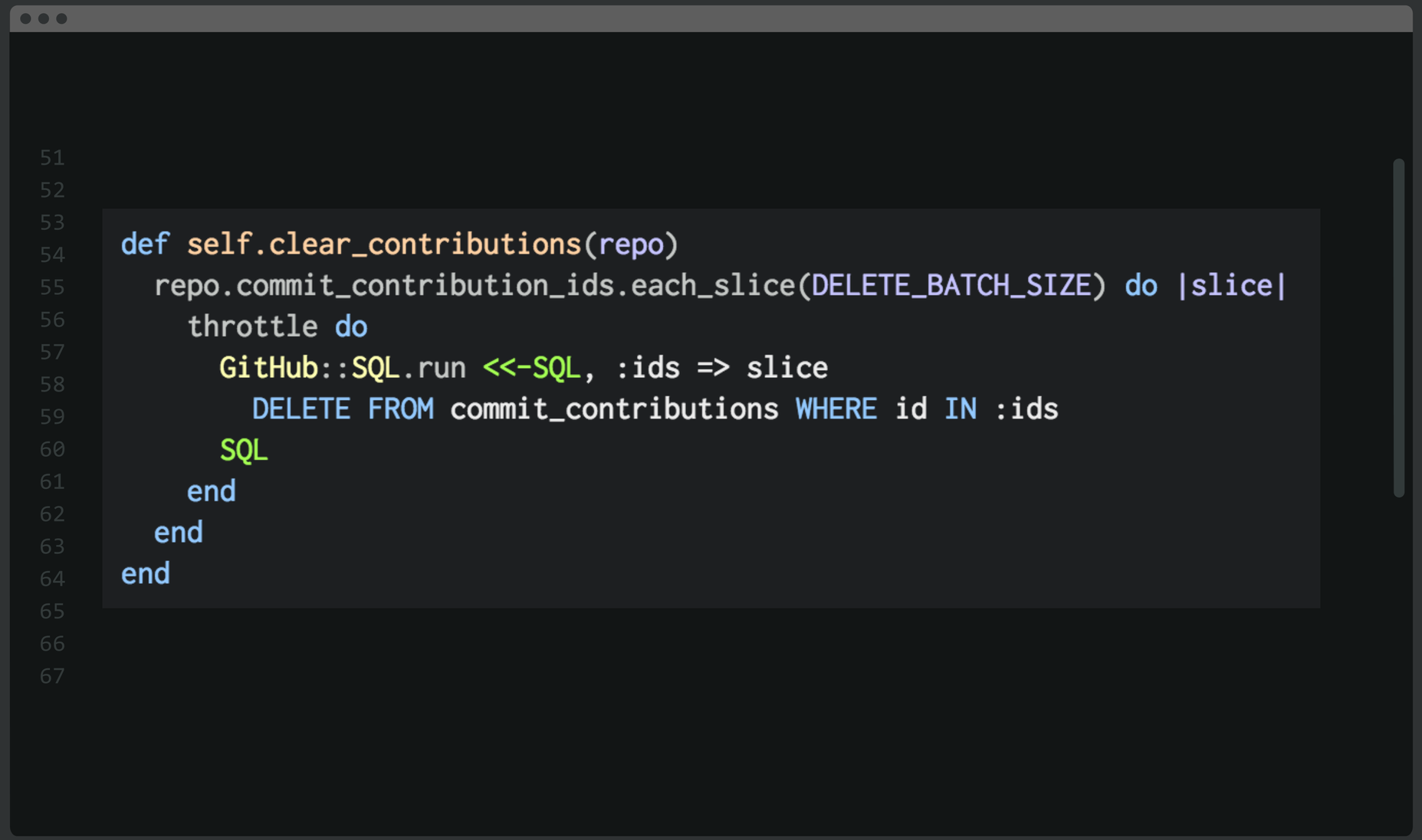
# Throttler

# Github::Throttler

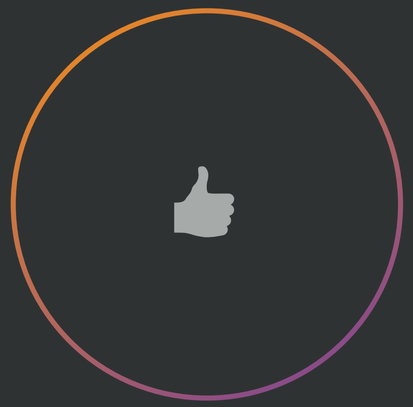


```
51  # Public: throttle based on replication delay
52  #
53  # Waits until replication delay is at or below ALLOWABLE_DELAY, then yields.
54  def throttle
55    GitHub.stats.increment "replication-delay.called"
56    slept = 0
57    total_slept = 0
58
59    loop do
60      begin
61        hostname, delay = maximum_replication_delay
62
63        if delay <= ALLOWABLE_DELAY
64          return yield
65        end
66
67        log "#{hostname} delay #{delay} > #{ALLOWABLE_DELAY}, sleeping"

          sleep ALLOWABLE_DELAY
          slept += ALLOWABLE_DELAY
          total_slept += ALLOWABLE_DELAY
          GitHub.stats.increment "replication-delay.slept"

          if slept >= REPLICA_RESET_TIME
            GitHub.stats.increment "replication-delay.reload-replicas"
            reset_replicas
            slept = 0
          end
```

# Github::Throttler



```
51
52
53  def self.clear_contributions(repo)
54    repo.commit_contribution_ids.each_slice(DELETE_BATCH_SIZE) do |slice|
55      throttle do
56        GitHub::SQL.run <<-SQL, :ids => slice
57          DELETE FROM commit_contributions WHERE id IN :ids
58        SQL
59      end
60    end
61  end
62
63
64
65
66
67
```



# Problem solved

We are hiring!!

git.io/vv8yi

# Roadmap

- GTIDs
- MySQL 5.7
- Cluster aware monitoring
- Failover improvements
- So much more

# Questions?

# Sam Lambert

**Director of Technology**

github.com/samlambert
twitter.com/isamlambert
samlambert.com