


Seeing the invisible: two years at Wikipedia with W3C's Network Error Logging

Chris Danis

 @cdanis@hachyderm.io

Who's Wikimedia?



WIKIPEDIA
The Free Encyclopedia



Wiktionary
The free dictionary



WIKIDATA



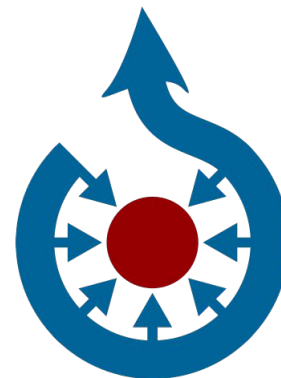
WIKIQUOTE



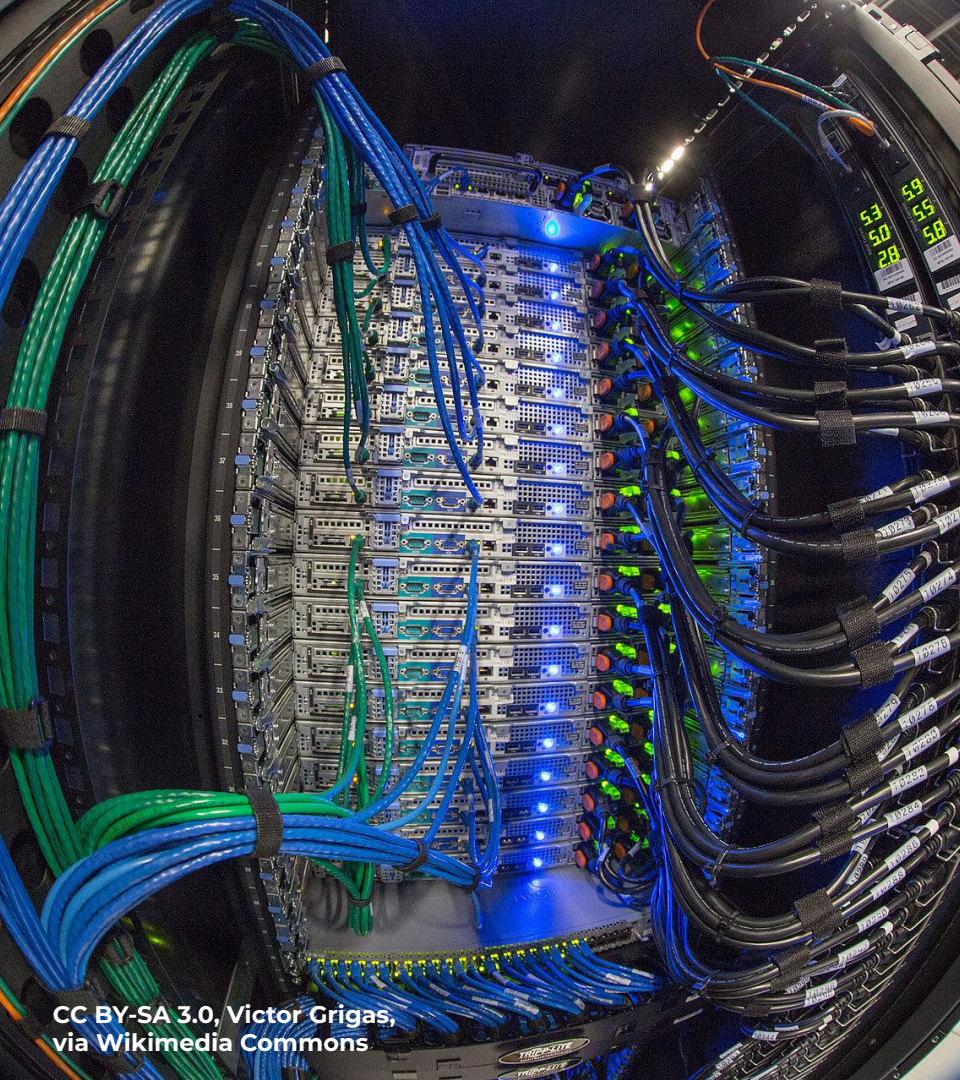
WIKIMEDIA



**wiki
voyage**



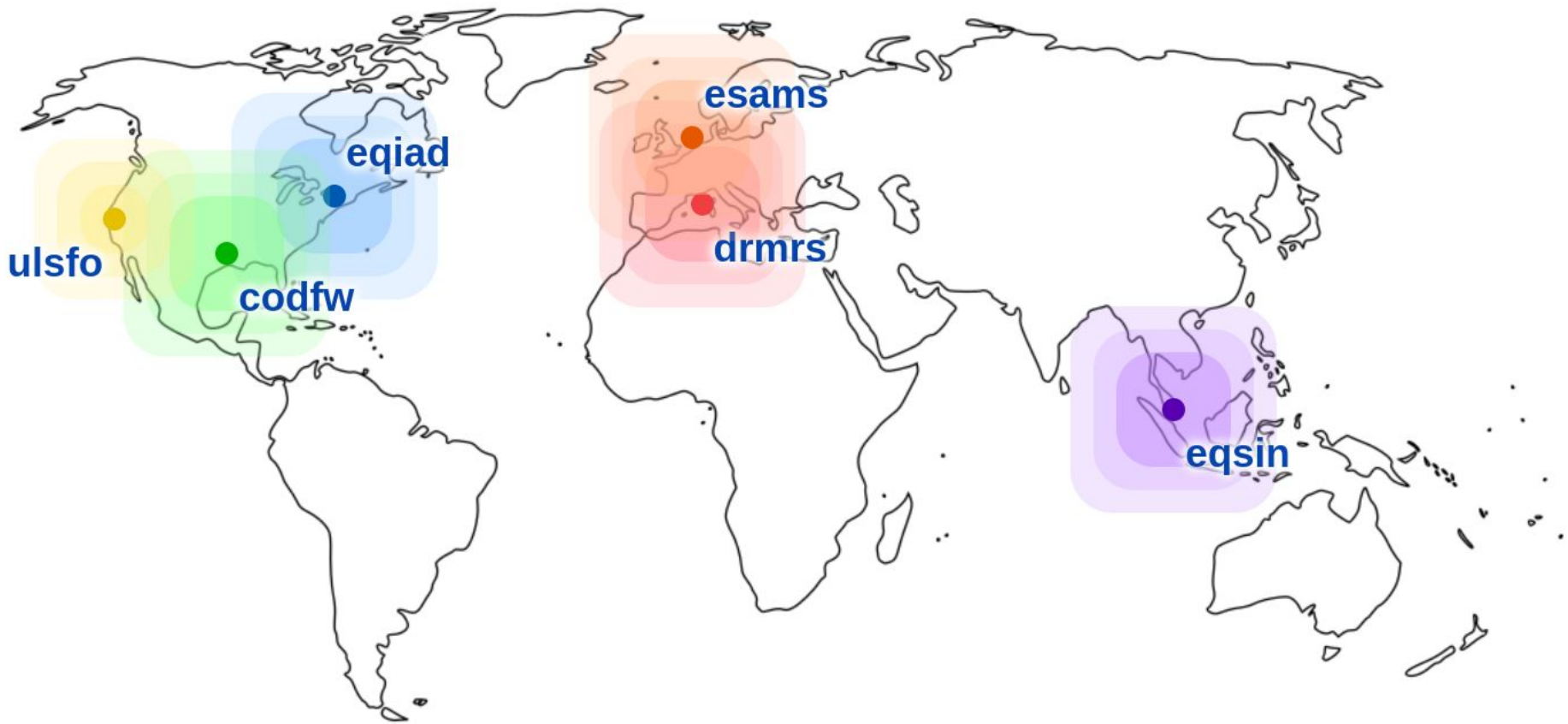
**WIKIMEDIA
COMMONS**



CC BY-SA 3.0, Victor Grigas,
via Wikimedia Commons

Our infra

- All our own metal
- ~1,800 physical servers
- 6 total locations
 - 2x "core" + CDN
 - 4x just CDN



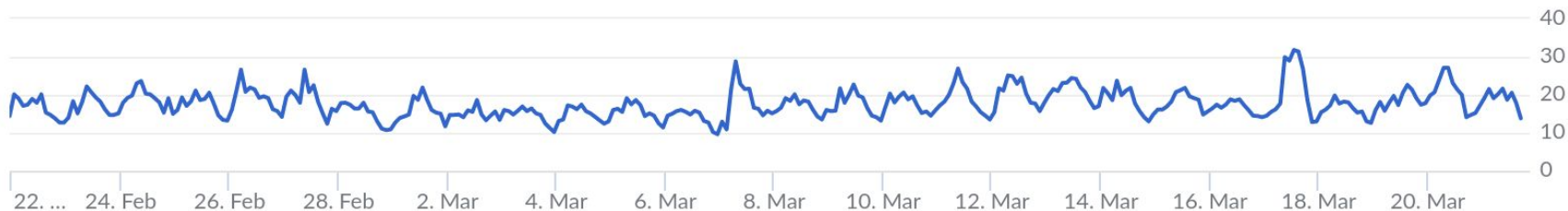
Total request volume ?

116,867 requests/second

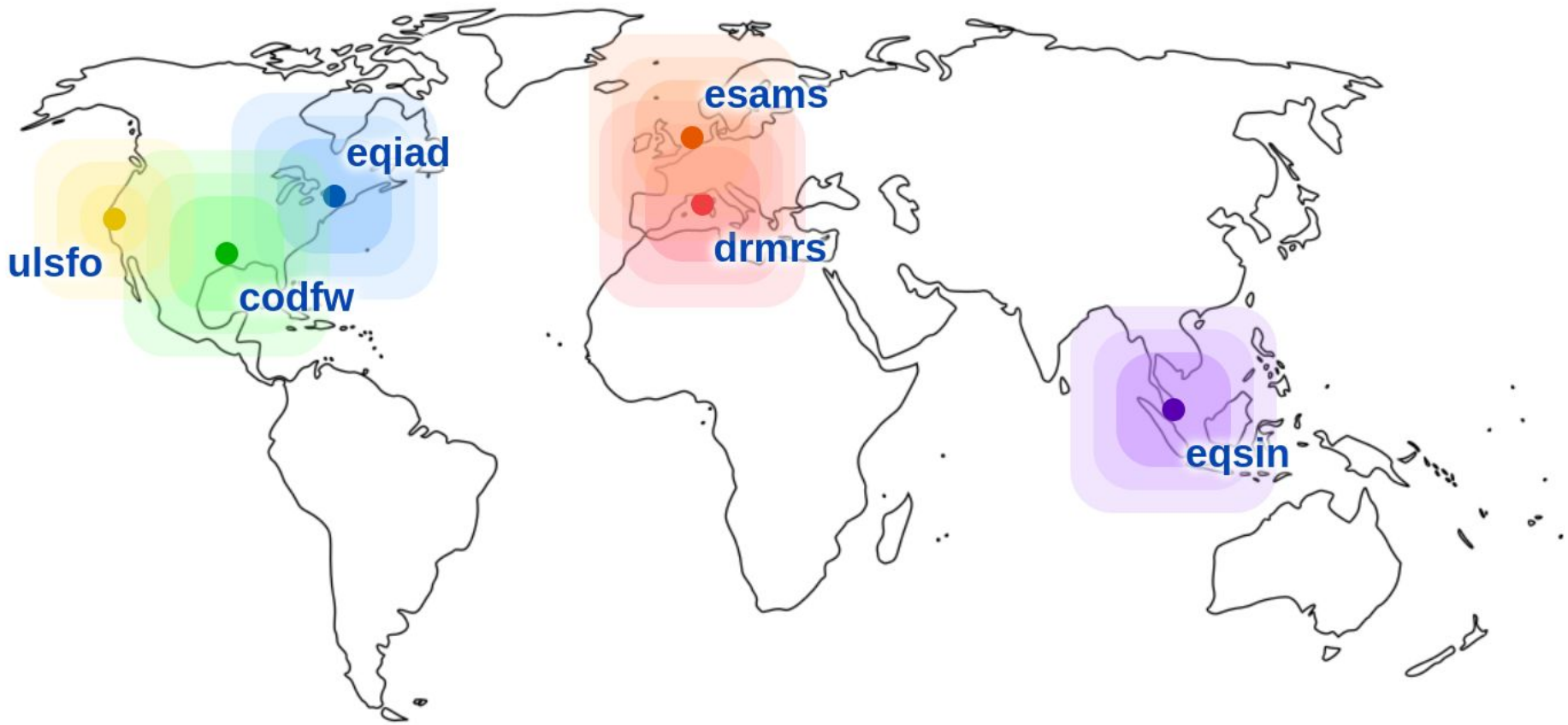


Successful edits ?

13.8 edits/second



— www.wikmediastatus.net (or <https://grafana.wikimedia.org> for even more)



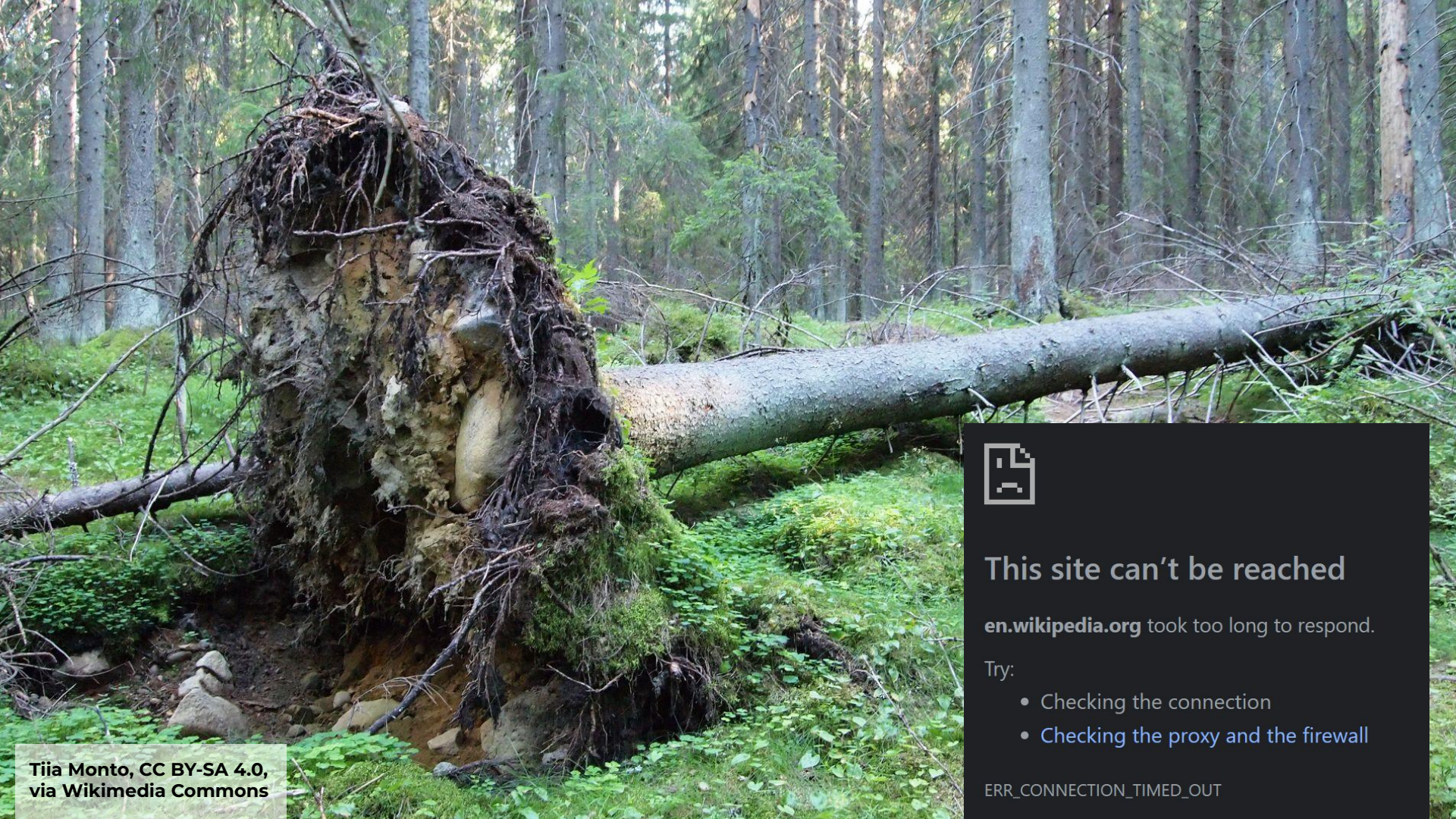
A large, fluffy white cloud is centered in the upper half of the frame, set against a clear, vibrant blue sky. The cloud has a soft, textured appearance with various shades of white and light blue. The overall composition is simple and clean.

“The Internet”

Dmitry Makeev, CC BY-SA 4.0,
via Wikimedia Commons



Agnes Monkelbaan, CC BY-SA 4.0,
via Wikimedia Commons



Tiia Monto, CC BY-SA 4.0,
via Wikimedia Commons



This site can't be reached

en.wikipedia.org took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_TIMED_OUT

















Yes.

@timestamp	Oct 28, 2021 @ 19:06:58.574
# age	0
# body.elapsed_time	30,313
t body.method	GET
t url	https://en.wikipedia.org/
t body.phase	connection
t body.protocol	http/1.1
# body.sampling_fraction	0.05
# body.status_code	0
t body.type	tcp.timed_out
IP http.client_ip	1xx.2yy.aaa.bbb
t isp_asn	Vodafone Italia DSL (AS 30722)
t http.request_headers.user-agent	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Sa

```
> GET / HTTP/2
> Host: en.wikipedia.org

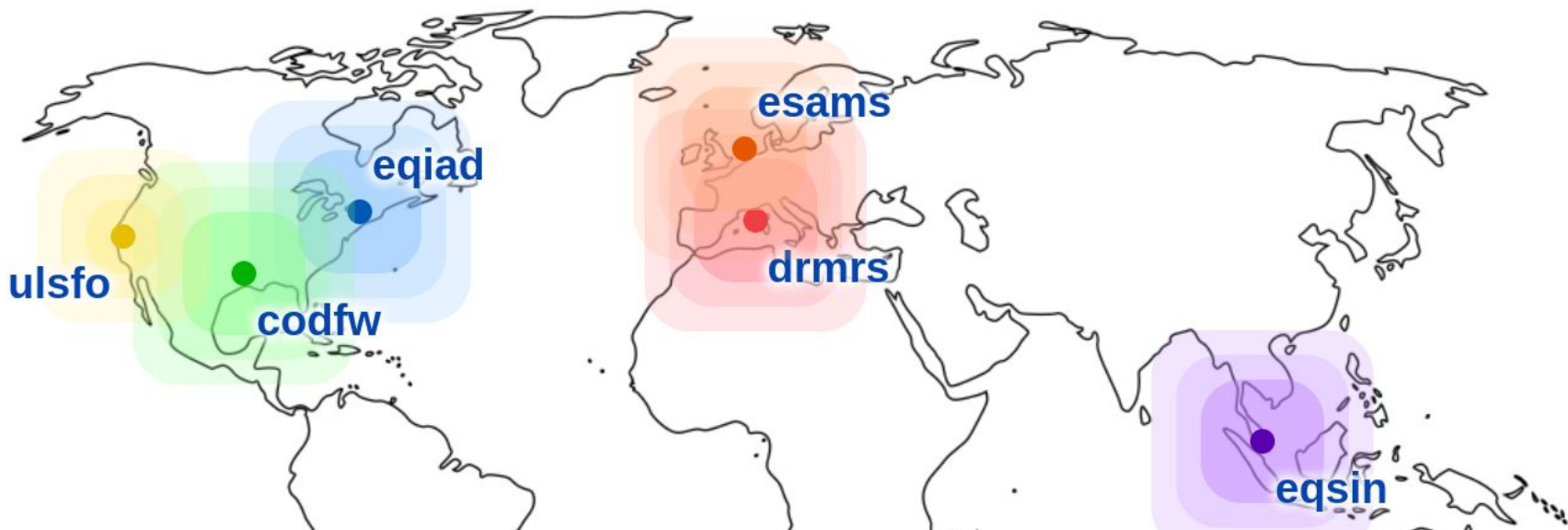
< HTTP/2 301
< location: https://en.wikipedia.org/wiki/Main_Page
< report-to:
  {
    "group": "wm_nel",
    "max_age": 86400,
    "endpoints": [{
      "url": "https://intake-logging.wikimedia.org/..."
    }]
  }
< nel: {
  "report_to": "wm_nel",
  "max_age": 86400,
  "failure_fraction": 0.05,
  "success_fraction": 0
}
< [...]
```

											
NEL 	<	<	✘	<	✘	<	✘	<	✘	<	<
	71	79	No	58	No	71	No	50	No	10.2	71
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 WebView Android



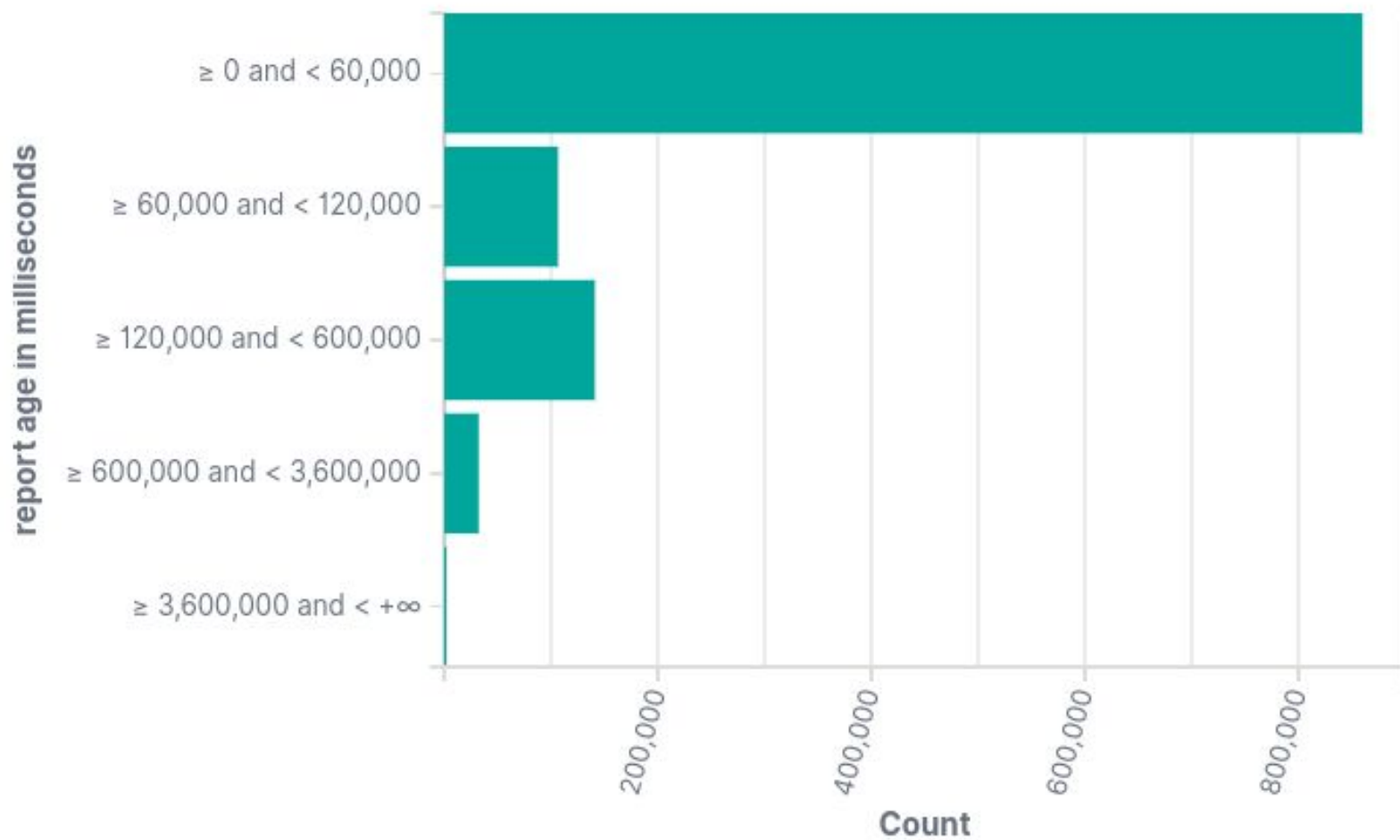
Yes.

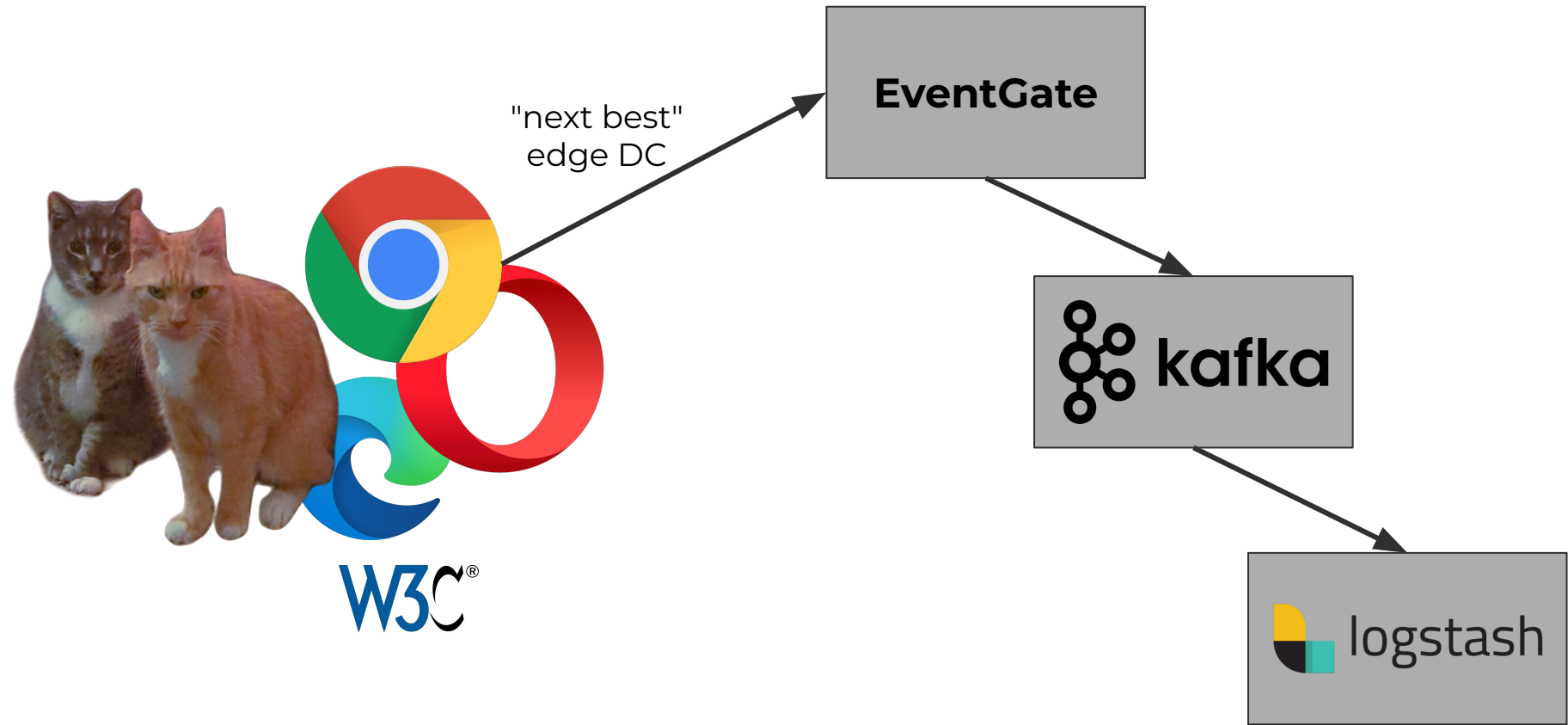
... 5% of the time
... eventually



```
✓ cdanis@mcnulty ~/work/gits/dns ☕ git grep -A3 text-next
templates/wikimedia.org:intake-logging 600 IN DYNA geoip!text-next
--
geo-resources:text-next => { # Returns next-best choice vs text-addr
                             # above, for NEL (T257527 & T261340)
geo-resources- skip_first => true # This flag is new in v3.3.0
```


NEL report age in milliseconds







Timeline

- Began work in July 2020
- Launched Sept 2020
- IP geolocation of reports added March 2021
- NEL-based paging starting Oct 2021

Anatomy of a NEL report

- URL
- HTTP method (GET, POST, HEAD, ...)
- Protocol (http/1.1 or http/2)
- Referrer
- User-Agent string



Anatomy of a NEL report

- `elapsed_time`: client-measured duration of the request (until error or success)
- `age`: how many milliseconds old the report was at the time it was uploaded
- The sampling fraction at generation time



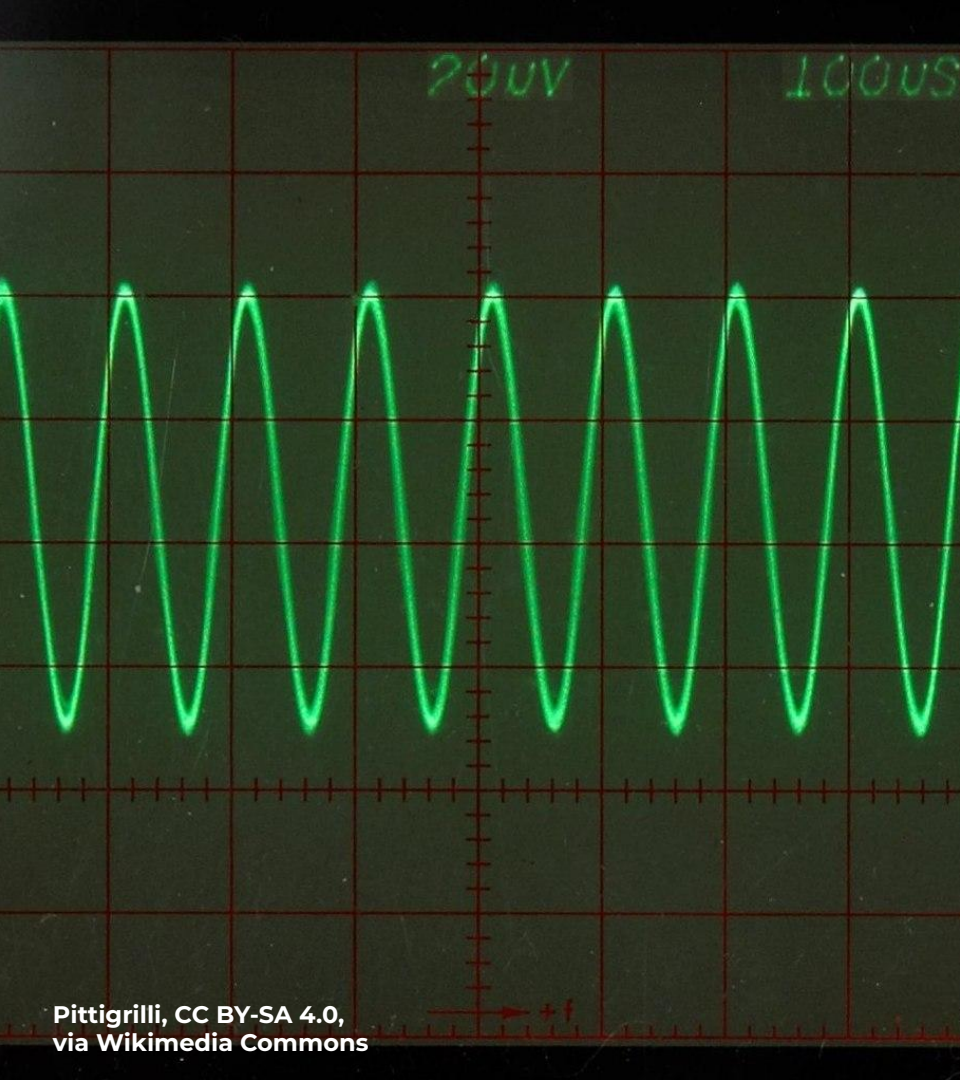
Anatomy of a NEL report

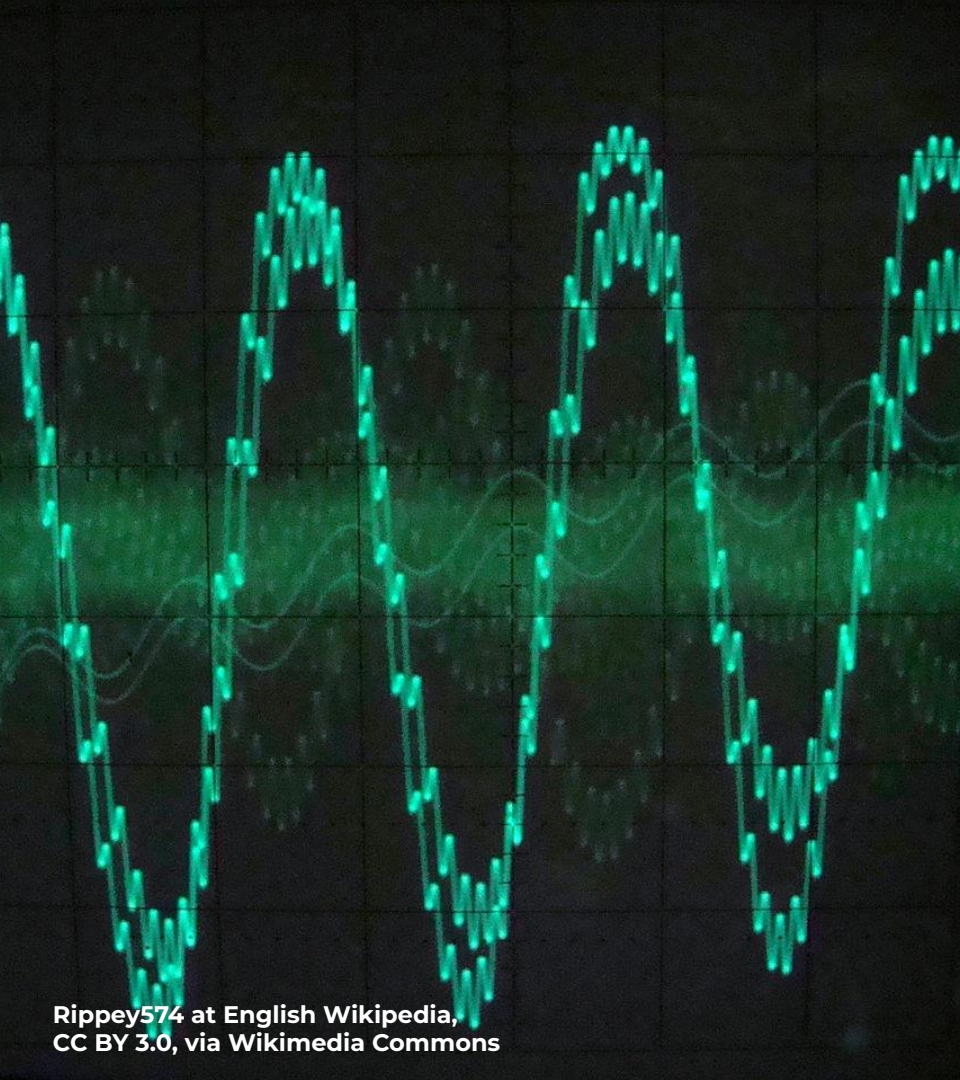
- HTTP response status code, if any
- phase: always one of `dns`, `connection`, or `application`
- type: partial list of examples:
 - ❑ `dns.name_not_resolved`
 - ❑ `tcp.timed_out`
 - ❑ `tls.cert.date_invalid`
 - ❑ `http.response.redirect_loop`



Highest signal

- `tcp.timed_out`
- `tcp.address_unreachable`
- `dns.name_not_resolved`





Ripsey574 at English Wikipedia,
CC BY 3.0, via Wikimedia Commons

Noisy signal

- `tcp.reset`, `tcp.closed`
- `h2.ping_failed`
- `abandoned`

Just noise

- `http.error`
- `unknown`

What's not in a NEL

- Client IP address
- Any derived-from-IP data, like a geolocation, AS number, etc.
- Timestamps



Time is relative

"Each report is delivered along with an **age** property, rather than the [local] timestamp at which it was generated. We do this because each user's local clock will be skewed from the clock on the server by an arbitrary amount. The difference between the time the report was generated and the time it was sent will be stable, regardless of clock skew, and we can *avoid the fingerprinting risk* of exposing the clock skew via this API."

– <https://www.w3.org/TR/reporting/#fingerprinting-clock-skew>



Incident case studies: #1

Corrupted PDF downloads

- User reports were our only indication of the problem
- All service metrics looked fine
- Could not reproduce with internal requests
- Inconsistent and intermittent, same file sometimes worked sometimes didn't, no obvious temporality
- At its worst, was affecting about ~20% of downloads





The onion of production

- The corruption was actually truncation
- Initial suspicions were around the PDF generator backend itself, or the reverse proxies nearest to it



The onion of production

- Turned out to be an issue in Varnish (in the middle of our CDN) with a parameter known as `nuke_limit`
- Full story at <https://phabricator.wikimedia.org/T266373>

Network Error Logging 107 hits

New Save Open Share < ~ 15 days ago to ~ 11 days ago >

url:"VapiVrest_v1VpageVpdf"

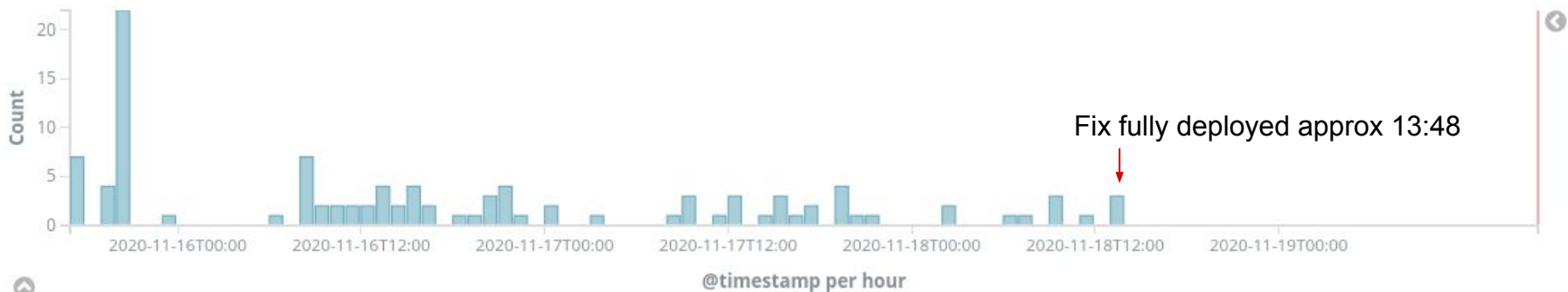
Uses lucene query syntax

report_body.type: "abandoned" report_body.status_code: "404" query: "report_body.type:unknown AND report_body.status_code:200"


meta.stream: "w3c.reportingapi.network_error" report_body.status_code: "200" report_body.type: "http.response.invalid.content_length_mismatch"

Add a filter + Actions

2020-11-15T16:57:39 - 2020-11-19T16:57:39 — Auto



Time	report_body.type	report_body.server_ip	url
2020-11-18T13:28:14	http.response.invalid.content_length_mismatch	91.198.174.192	https://en.wikipedia.org/api/rest_v1/page/pdf/The_Lion%2C_the_Witch_and_the_Wardrobe

 **BBlack** added a comment. Nov 18 2020, 21:18 

No reports of the PDF truncations in NEL for ~8 hours now, which is a significant break from recent trends. Can anyone else still repro this in any way?

 **RhinosF1** added a comment. Nov 18 2020, 21:25 

Works for me

— <https://phabricator.wikimedia.org/T266373>

Incident case studies: #2



Call before you dig

*Dear Valued Customer,
Repair crew is still working to expose
conduit, they are breaking the
concrete and slurry away from
around the duct structure ... Techs in
the field are reporting that there are
no spare ducts available ... There are
5-6 other contractors in the area
trying to repair their ducts as well so
progress is pretty slow at the
moment... No ETR at this time.*

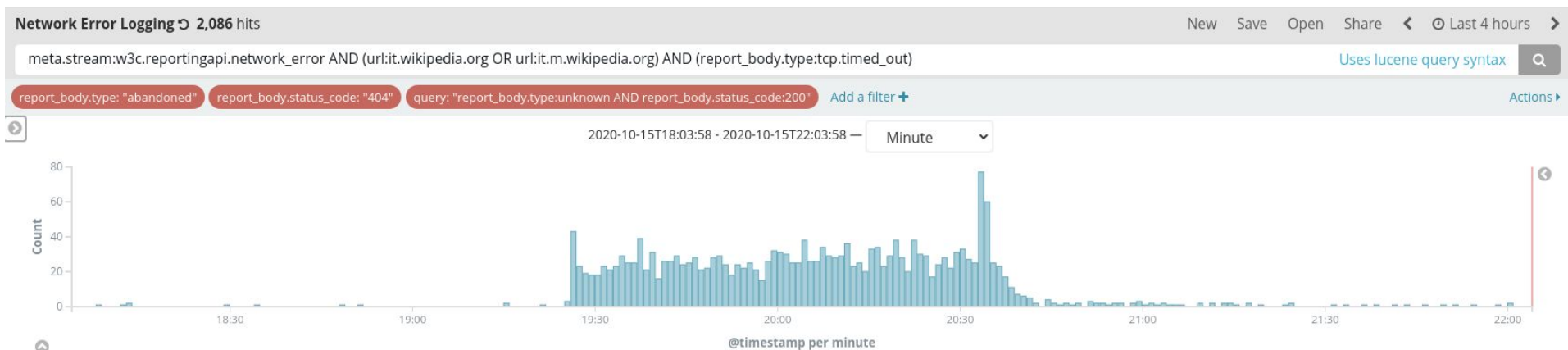
User-reported network connectivity errors (NEL)



Incident case studies: #3

Too much scrubbing

- Use a few different L3/L4 DDoS "scrubbing" services
- "Well, there's your problem"



Otherwise (if *result* is "Failure"):

1. Increment *endpoint's* failures.
2. Set *endpoint's* retry_after to a point in the future which the user agent chooses.

Note: We don't specify a particular algorithm here, but user agents are encouraged to employ some sort of exponential backoff algorithm which increases the retry period with the number of failures, with the addition of some random jitter to ensure that temporary failures don't lead to a crush of reports all being retried on the same schedule.

ISSUE 2 Add in a reasonable reference describing a good algorithm. Wikipedia, if nothing else.

– <https://www.w3.org/TR/reporting/>



Closing thoughts

Future work @ Wikimedia

- NELs as part of an end-to-end SLO for our CDN
- Mapping real user latency to each of our PoPs using NEL `success_fraction`



Off the shelf

- At least one CDN now has an "enable NEL" toggle switch
- Several uptime monitoring solutions offer NEL collection



DIY

- Receive a JSON array over HTTP POST
- Do some basic aggregation and analysis, perhaps also IP geolocation
- Set some HTTP response headers in your app
- Have some way of running this outside your usual request path (at least for any given user)





But a few caveats

- Pick your TTLs and sampling fractions wisely
- Reports can be recursive
- Be prepared for background noise — understand it in advance, don't get lost in it

Either way...

- Caveats aside, it's not hard to set up
- Gives you end-to-end observability of real user experiences
- If you run a CDN or use someone else's:
Why not NEL?




Further reading & resources

- https://wikitech.wikimedia.org/wiki/Network_Error_Logging and <https://phabricator.wikimedia.org/T257527> and <https://codesearch.wmcloud.org/operations/?q=T257527>
- <https://www.wikimediastatus.net/> and <https://grafana.wikimedia.org/>
- <https://www.w3.org/TR/network-error-logging/>
- <https://www.usenix.org/conference/nsdi20/presentation/burnett>





Chris Danis

 @cdanis@hachyderm.io