Page   Discussion   Read   View source   View history   Search Wikitech   🔍

Toolforge webservices are in the final stages of   migrating to the toolforge.org domain .
Please help us clean up older documentation referring to tools.wmflabs.org!

Incident documentation/20190301-proton

< Incident documentation

<div>

Contents [hide]

</div>

Summary

Proton, one of the two currently deployed PDF rendering solutions in WMF, received an increased amount of requests due to a bot for a period of 2 hours, which it was unable to handle, leading to a degraded service for parts of that period

Timeline

All times are UTC

11:00 traffic increases for proton by a factor of 3. However nothing bad happens as the infrastructure is able to handle it

11:39 traffic increases even more, now by a factor of 6. During the next 1.5h the traffic will gradually increase even more, reaching up to 25req/s for the a4 desktop endpoint, an increase of over 25x. A graph can be seen at https://grafana.wikimedia.org/d/000000563/proton?orgId=1&from=1551437310272&to=1551448110272 . Logstash at https://logstash.wikimedia.org/goto/524c3c5e029d90bb9a84ced759540f79 pointing out the the queue is full

12:04 traffic has reached a peak.

12:04 icinga-wm alerts, but the service recovers quickly leading to believe it's a false positive

12:22 icinga-wm alerts again, alex starts looking into it

12:23 alex reports finding 32 chromium processes on 1 of the 2 proton hosts in eqiad and another 99 chromium processes on the other one. Some have dates of Jan 30.

12:31 akosiaris: restart proton on proton1001 [1]

12:32 OOM locks up proton1002. The service keeps on flapping. [2]

12:35 alex validates theory that some 3rd party is creating a lot of pdfs

12:38 alex realizes some 3rd party is trying to create pdfs of many dewiki articles

12:41 icinga-wm sends the first SMS pages, recovers quickly

12:44 alex silences the SMS alerts for the next 2 hours, to avoid unnecessarily spamming the rest of the team while investigation is ongoing

12:49 after a discussion with ariel: akosiaris: lowers the max_render_queue_size [3]

12:50 Giuseppe suggests varnish rate limiting just for the PDF endpoint of the API

13:00 alex states the requests are for restbase aggregated PDF endpoint API.

13:02 alex lowers the limits for both concurrent rendering and queue to 1 and 10 respectively.

13:05 alex after digging through logs realizes that an external IP in the EU has requested over 36K PDF versions of articles in the span of 2 hours, which back of the envelope calculations amount to ~5req/s, which is on average below the current limit of 10req/s

13:08 Ariel suggests we look at the User-Agent

13:09 alex reports that accept-language, user-agent and referrer headers are all null, making it clear it's a bot talking to the API directly

13:18 Emanuelle merges a patch limit the API to 1req/s at the edge cache level

13:20 the bot has already stopped

14:00 the bot presumably tried to restart but was rate limited successfully at the edge cache level

15:02 alex restores the proton config values

Conclusions

The infrastructure powering the proton API was not sufficient to address an unexpected 25x increase in the amount of requests. We did not have proper rate limit for a particularly expensive API endpoint. If the service was on the kubernetes infrastructure it would have been possible to increase the resources for the service, something not possible currently.

Links to relevant documentation

- runbook: None currently exists

Actionables

- Migrate the proton service to kubernetes (TODO: Create task)
- Create a runbook for issues with the proton service (TODO: Create task)
- Add PDF API specific rate-limiting per IP ✔ **Done**
- Investigate 2019-03-01 Proton incident - phab:T217724

Category: Incident documentation

This page was last edited on 6 March 2019, at 08:40.

WIKIMEDIA project   Powered By MediaWiki