Create account Log in

Page Discussion Read View source View history Search Wikitech

Main page
Recent changes
Server admin log (Prod)
Server admin log (RelEng)
Deployments
SRE/Operations Help
Incident status

Cloud VPS & Toolforge
Cloud VPS documentation
Toolforge documentation
Request Cloud VPS project
Server admin log (Cloud VPS)

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page

Print/export
Create a book
Download as PDF
Printable version

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/20170106-Cache-upload

< Incident documentation

From 16:08 to 18:11 UTC on Friday, 06 January 2017 several requests for images and other multimedia files resulted in user-facing 503 error messages. The root cause of this outage was an administrative command being executed on multiple cache servers in a data center disabled earlier during the morning due to network issues (codfw). Because of an operator mistake, the command was executed on all machines at the same time rather than in a rolling fashion, which triggered a bug in the caching proxies (varnish) on the majority of our CDN nodes, including those in other data centers. When a caching proxy process crashes it gets restarted, and its cache contents emptied. The origin servers responsible for serving multimedia content thus became overloaded and started misbehaving. In particular, all swift frontends showed high load, as well as the backends responsible for accounts: ms-be1002, ms-be1005, and ms-be1022. The issue was solved by directing requests for image thumbnails to the secondary data center, thus relieving pressure from the overloaded origin servers in the primary DC.

Timeline

All times UTC.

- 16:05 Administrative command executed
- 16:08 Outage begins
- 16:12 First page received
- 16:17 bounce swift-proxy on ms-fe100[123] leave ms-fe1004 for investigation
- 17:12 root cause identified (caching proxies crashes)
- 18:08 switch thumbs to codfw
- 18:12 all requests back to normal

Conclusions

Depooling and repooling cache nodes has important consequences. It should be done in sequential order even on depooled data centers. Varnish reloads should not be considered 100%-safe operations, which is something we already noticed in the past.

A dashboard plotting the number of objects currently in cache, as well as alerting on crashing varnish processes, would have helped speeding up root-cause identification.

Pressure was relieved from swift in eqiad by routing requests for thumbnails to codfw. A knob allowing to choose which percentage of traffic should go to which data centre could be an interesting idea for the future.

On the swift side, the 3 backend servers responsible for accounts have been identified as one the bottlenecks. The swift replication factor for accounts should be increased to allow for more fan-out. Furthermore, we have noticed swift proxy had their CPU saturated during the time of the incident. This is likely due to swift frontends in eqiad having Hyper-threading disabled, it should be enabled across the swift cluster.

Actionables

- Further investigate varnishd crashes Task T154801
- Plot number of cached objects on per-server / per-cluster grafana dashboard Task T154864
- Increase swift replication factor for accounts Task T156136
- Possibly alert on varnishd crashes (TBD)

Category: Incident documentation

This page was last edited on 24 January 2017, at 14:52.

details.