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

Incident documentation/20180731-Phabricator

< Incident documentation

Contents [hide]

Summary

On 31st July 2018, Wikimedia's Phabricator instance (https://phabricator.wikimedia.org) was instable, showing non-canonical data, unavailable or in read only mode for about 10 minutes, and within those, also lost some ticket information for a period of about 6 minutes.

Timeline

15:02: Started network maintenance to move servers on EQIAD row B to a different switch Task T183585

15:18: dbproxy1008 and dbproxy1003 detect db1072 as down (this host was part of the network maintenance and was expected to have a small downtime)

- At that moment some connections (both reads and writes) started going to db1117:3323 (which is supposed to be on READ ONLY), while others continue writing to db1072
- DBAs start investigating

15:18: `<icinga-wm> PROBLEM - MariaDB Slave SQL: m3 on db1117 is CRITICAL: CRITICAL slave_sql_state Slave_SQL_Running: No, Errno: 1062, Errmsg: Error Duplicate entry 24361579 for key PRIMARY on query. Default database: phabricator_file.`

- Icinga also alerts that there are duplicate errors, creating a split-brain scenario

15:22: DBAs realize db1117:3323 wasn't in read only, being the cause of the replca writes and the broken replication.

15:24: db1117:3323 is set to read only while research continues

- At this point phabricator starts being unavailable, because database read-only mode is not supported
- A decision has to be made if to continue letting db1117:3323 as the master, or reverting back to db1072. Both options imply a small, but indetermined amount of data loss. db1072 is decided on grounds that it will provide higher availability (as it will allow codfw replicas be an exact replica)

15:26: DBAs reload haproxy to make sure db1072 is back as being the master.

15:27: Due to phabricator connection pool, connections already established to db1117:3323 would need to be killed just to make sure - to stop MySQL is decided as it is the safest option

15:28: Phabricator is confirmed to be fully back up

Conclusions

A series of events triggered this event.

- dbproxy1003/8 were not handled/prepared appropriately for the scheduled network maintenance, as they were not direct part of the maintenance (the most they monitor were!). A special reminder has to be set in the future to test haproxy hosts being potentially involved.
- A downtime a bit bigger than expected (haproxy at the time considered a host failed after 9 seconds (3000ms * 3 retries) and bigger than the usual other net maintenances that have been done. Previous network maintenance hadn't caused an automatic failover
- Phabricator uses heavy connection pooling, which affects negatively the time it takes for the failover to

happen (resulting on only some writes being diverted, resulting on the actual split brain)
- It was thought that the failed to host was in read_only, preventing accidental split brain. This wasn't true, an incorrect read_only configuration on the MySQL replicas for misc made the split brain possible, as the replica was writable. This was due to a mistake on the relatively new role puppet manifests (misc_multiinstance), due to, at the time, the 2 methods to configure read_only: on the template and as a config parameter.

Actionables

- Status: **Done** - Fix puppet code so the appropriate hosts are in read_only mode gerrit:450205
- Status: **Done** - Monitor read only variables on replicas and alert (on IRC) if, in the future, replicas are writable phab:T172489
- Status: **Done** - Increase the timeout, from 9 seconds to 60 seconds, for haproxy to consider a host hard down gerrit:450542
- Status: **Pending** - (Not strictly related to this) haproxy is lacking proper logging on dbproxy roles (or at least, dbproxy1002) phab:T201021

Category: Incident documentation