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/20160705-commons-replication

< Incident documentation

Contents [hide]

Summary

All Commons (s4 shard) slaves failed its replication, setting the service as read-only between 10:22 and 10:45, until its master was failed over. Commons was unavailable for editing for that period. For a brief period after that, both recentchanges and api were also outdated, causing some issue such as frozen recentchanges and gadget malfunctions. As commons is a project with shared information for other wikis, some administrative actions (such as deletions) on other wikis were also temporarily affected.

Timeline

- Server admin log entries during this time
- **10:22**: a deletion action `LocalFileDeleteBatch::doDBInserts` is sent to s4-master. All slaves failed to replicate that action due to a duplicate key error: `Error Duplicate entry 3983277 for key PRIMARY on query. Default database: commonswiki. Query: INSERT INTO filearchive...`
- **10:27**: icinga notifies both on chat and SMS of the issue
- **10:31**: the alerts and servers are evaluated and a global failure (master corruption or global drift) is identified as the immediate cause. Communication is sent that commons will be read-only for a while.
- **10:32**: master failover change is agreed
- **10:38**: the master failover finish deployment on all mediawiki servers. Edits are allowed again, but they will fail.
- **10:43**: icinga confirms replication recovery on all slaves.
- **10:45**: new master is put back in read-write mode. Actual successful edits start again.
- **10:47**: icinga announces failures again
- **10:49**: All slaves except db1081, db1084 and db1091 are stopped again (`table commonswiki.filearchive cannot be converted from type varchar(100) to type varbinary(32)`). This affects mainly API and recentchanges visualization.
- **11:00**: recentchanges is failed over to db1081
- **11:09**: issues with api still reported
- **11:15**: api and all other roles also failed over to db1081. Api issues finish here.
- **minutes later**: other non-directly-user facing issues such as icinga bogus alerts, puppet changes, tendril/dbtree are fixed in subsequent commits (outside of the outage period).

Conclusions

- There was slave drift on the database. While that is always a possibility, this caused even worse issues because it affected all slaves, so we have to conclude that it is the master at the time (db1042) what originated the issue. There is 2 possibilities for that: an "unsafe" query was executed at some point, that has different effects when executed serially than on the slave (something that could not be discarded, given that we use STATEMENT based replication, unsafe_statements_for_binlog and relatively unsafe queries) or the slave was promoted, already containing that drift beforehand (because of an operational error, or the previous

reason).

- The outage was extended when a second replication error happened- this second error was caused by the combination of a partially applied schema change (only applied to db1040, and because they where cloned from it, db1081, db1084 and db1091) + misconfiguration of not changing binlog_format to STATEMENT on the new master (it is MIXED on the slaves).
- The failover was done easily thanks to GTID replication, which allowed very easy failover between hosts. That is a positive point.
- On the negative point, a query had to be executed for each server + a couple of code deployes, delaying the failover, which otherwise could be done on a single script automatically. This increased the outage time by ~3 minutes.
- On the positive side, having more servers/capacity than usual available allowed an easy failover, and cloning them from the same server (previously available master) was a good decision
- Failover procedure needs to be more agile and not depend on code deploy, with either bug T24923 or the introduction of etcd-controlled pooling. This increased the outage time by ~8-10 minutes.
- Until bug T104459 and bug T108255 or bug T109179 are completed, this will likely happen again

Actionables

- Status: **Unresolved** Fix pending issues related to the failover - revert weight changes (bug T139346)
- Status: **Unresolved** Automatize checking and fixing of object, schema and data drifts between production masters and slaves (bug T104459)

Category: Incident documentation