Create account Log in

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/20160601-MediaWiki

< Incident documentation

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

Contents [hide]

Summary

Mediawiki servers served HTTP 5xx for 101 seconds during the 2016-06-01 SWAT window.

This incident is a consequence of the current state of our configuration testing infrastructure and deployment processes.

A constructive brainstorming session occurred on #wikimedia-operations to identify clear actionables to harden up our deployment workflow.

Timeline

Timeline is UTC.

Before the incident

- 23:00 The 2016-06-01 SWAT window starts, this is a busy SWAT with config changes and wmf4 changes.
- 23:45 MatmaRex noticed an incident, not related to the SWAT, about CentralNotice undeployed change. This will create a small source of distraction during the SWAT.

Incident

- 23:49 A patch is merged, authored by Dereckson, CR+1 by Luke, to migrate SpamBlacklist to extension registration (gerrit:281239). This patch contained a typo: SpamBlack**L**ist instead of SpamBlack**l**ist. This typo is easy to spot by automated systems but hard to spot to humans, as demonstrated by the CR+1.
- 23:49:49 dereckson@tin Synchronized wmf-config/CommonSettings.php: Use extension registration for SpamBlacklist (T119117) (duration: 00m 24s). At this moment, the incident starts and a spike of 5xx requests appear.
- 23:5x:xx Test shows `Fatal error in MediaWiki servers: PHP fatal error /srv/mediawiki/php-1.28.0-wmf.3/includes/GlobalFunctions.php line 115: /srv/mediawiki/php-1.28.0-wmf.3/extensions/SpamBlackList/extension.json does not exist`, the change is immediately reverted.
- 23:51:30 dereckson@tin Synchronized wmf-config/CommonSettings.php: Revert Use extension registration for SpamBlacklist (T119117) (duration: 00m 24s) — At this moment, the incident is solved.

After the incident

- 23:56:34 The revert commit was merged into the operations/mediawiki-config repository afterwards.
- 00:00:59 The SWAT is stopped, on Ori suggestion, to discuss the situation and see what we could do.
- 00:50 We resumed the SWAT to deploy the merged undeployed changes, with a full mw1017 testing before sync to prod.
- 01:25 End of the SWAT. At this moment, code merged in wmf branches = code deployed on the server.

Conclusions

What weakness did we learn about and how can we address them? This incident, similar to 20160407-Mediawiki and more recently 20160212-AllWikisOutage, shows the current deployment process isn't bulletproof: humans errors in configuration (or sometimes code) can lead to break mwxxxx servers in production.

We should focus on offering ways to avoid such human errors and ease testing before the change reaches production. There are existing tools and methods already in place to avoid them. With minimal efforts, we could leverage them to offer a sensibly more solid deployment process.

Reinforce the efficiency of deploy process

The incident could have been avoided if deployed on mw1017 first. Previous incident reports already note that. The tricky issue is we don't have a time machine to know when we should have used mw1017 for a change. Yet, a `scap pull` on mw1017 is quick, a dozens of seconds at most. This swift time allows us to automate the mw1017 sync into the scap process.

To automate this auto sync to mw1017 feature will allow to leverage existing code to avoid most of such incidents. As it will be smooth and fully integrated in the process, it will be used for every change, not only changes especially identified for breakage risks. bug T110068#2347859 suggests a way to do this (see also Task T136839 "MediaWiki simple canary checks on mw1017").

This could also benefit from extra features to help to detect issues, some currently already planned, like bug T110068 — allow to get input from Logstash or Graphite.

As we've a clear vision and concrete actionables to solve these issues, it could be valuable to prioritize these deployment problems and assign resources to them. That will offer us a basic canary process.

Regarding "infrastructure" config changes

Some config changes add/set/remove values to customize a configuration for a wiki, generally on InitialiseSettings.php. They are fully suitable for SWAT and have a track record to be virtually without incident.

But some config changes, generally touching CommonSettings.php, should be more carefully processed.

Currently, these changes reach the SWAT process. Instead, we could provide a process to encourage deployers and change authors to request together a window in the Deployments calendar dedicated to their config infrastructure change instead of using SWAT for that. That would allow SWAT windows to focus on low risk changes.

Regardless of when it's deployed, a rule **Strongly consider to systematically use mw1017 for CommonSettings.php config changes** is valuable.

Allow better testing of config changes at development time

This incident was caused by a mere typo. This especially frustrating, as the conceptualized as catchable by verifications (unit tests, tests on beta cluster) prior to deployment: once live on any test server, the simple fact to request a page would have revealed the issue.

Our beta infrastructure doesn't allow easily to test that: we're in a queer workflow where a Jenkins job pick changes from the master branch (ie the production config) to apply them on the beta cluster. This seems strange and counterintuitive: it's reasonable to expect to be able to use the beta cluster to test that at development time, not at deploy time.

We should so offer an easy way to allow to test a Gerrit change, or a series of Gerrit changes, on a part of the beta cluster. Even the ones touching wmf-config/CommonSettings.php and not -labs.php.

Middle term: Prioritize QA

Deployment rate increased these last months, but quality assurance is insufficient and lagging. Developers should more take QA in consideration in their work duties.

Longer term: 1% production testing

During the 101 seconds of the incident, we served 100 to 150 000 5xx error requests. If we have a process to deploy to 1% of the production, that would have been limited to 1000-1500 error requests.

Such rolling deploy with intermediate levels of traffic check would be especially valuable for the MediaWiki train.

On a more positive note: what worked

This incident was swiftly solved. The fast-forward only scheme for the operations/mediawiki-config repository allow an immediate and straightforward `git revert` followed by a `scap sync-file` without any opportunity to mess the revert with the -m option. That probably contribute to decrease the downtime from 5 minutes to previous similar incidents to hundred seconds here. This is a gain of 67% in time.

Actionables

Explicit next steps to prevent this from happening again as much as possible, with Phabricator tasks linked for every step.

- Status: 🟥 **Unresolved** Automate checks against local MediaWiki from deployment host before syncing. (bug T121597)
- Status: 🟥 **Unresolved** Unit testing for operations/mediawiki-config repository: such things like case typo are hard to see by humans but easy to detect by tests. (bug T136782)
- Status: ⬜ **Declined** Investigate if wfLoadExtension couldn't caught exceptions to avoid a fatal error. When a fatal error is needed because an extension must absolutely be loaded for security reasons, assertions for the presence of this critical security extension could be made afterwards. (bug T136821)
- Status: 🟥 **Unresolved** Allow to test a mediawiki-config change to the beta cluster (bug T136828)
- Status: 🟥 **Unresolved** Automate mw1017 sync (Task T136839)
- …

Acknowledgement

Have participated to the the discussion following the incident, providing ideas and feedback included in this incident report: bblack, bd808, Debra (MZMcBride), Dereckson, gwicke, legotkm, MatmaRex, mutante, ori, RoanKattouw, twentyafterfour. This report summarizes inputs from #wikimedia-operations 2016-06-02 00:00 discussion.

Category: Incident documentation