



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/20190507-opcache

[< Incident documentation](#)

## Contents [\[hide\]](#)

- 1 [Summary](#)
  - 1.1 [Impact](#)
  - 1.2 [Detection](#)
- 2 [Timeline](#)
- 3 [Conclusions](#)
  - 3.1 [What went well?](#)
  - 3.2 [What went poorly?](#)
  - 3.3 [Where did we get lucky?](#)
- 4 [Links to relevant documentation](#)
- 5 [Actionables](#)

## Summary

Starting soon after a routine mediawiki config push to depool a db server, three different appservers each started serving three different nonsense-looking errors:

- mw1320: ConfigException from line 53 of /srv/mediawiki/php-1.34.0-wmf.3/includes/config/GlobalVarConfig.php: GlobalVarConfig::get: undefined option: 'UseKeyHe`der'
- mw1256: Error from line 578 of /srv/mediawiki/php-1.34.0-wmf.3/includes/libs/rdbms/database/Database.php: Undefined class constant 'DBO\_NOBUFFER'
- mw1271: Error from line 92 of /srv/mediawiki/php-1.34.0-wmf.3/vendor/wikibase/data-model-serialization/src/Deserializers/StatementDeserializer.php: Call to undefined method Wikibase\DataModel\Deserializers\StatementDeserializer:

The first one of these is an obviously-bogus string. The second one of these is an undefined class constant that has been in the MW code since 2016. The third is similarly an undefined method that has been in the code since 2016.

The believed cause is opcache corruption in PHP7 triggered by race conditions internal to PHP after `scap`'s invalidation of the opcache

## Impact

Approx. 8000 exceptions/fatal errors thrown over the course of approx. half an hour

## Detection

Icinga alert on 'MediaWiki exceptions and fatals per minute'

## Timeline

- 13:45: marostegui depools db1093, `scap sync-file db-eqiad.php`. Some hosts corrupt their opcache **OUTAGE BEGINS**
- 13:47: Icinga reports: PROBLEM - PHP7 rendering on mw1256 is CRITICAL: HTTP CRITICAL: HTTP/1.1 500 Internal Server Error - 539 bytes in 0.041 second response time
- 13:48: Icinga reports: PROBLEM - PHP7 rendering on mw1320 is CRITICAL: HTTP CRITICAL: HTTP/1.1 500 Internal Server Error - 106015 bytes in 0.181 second response time
- 13:55: Icinga reports: PROBLEM - MediaWiki exceptions and fatals per minute on graphite1004 is CRITICAL: CRITICAL: 90.00% of data above the critical threshold [50.0]
- 14:09: cdanis depools mw1320
- 14:12: cdanis depools mw1271, mw1256 **OUTAGE ENDS**
- 14:24: `_joe_` destroys php7 opcache on mw1320: `php7adm /opcache-free`

[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](#)

- 14:26: Icinga reports: RECOVERY - PHP7 rendering on mw1320 is OK: HTTP OK: HTTP/1.1 200 OK - 75404 bytes in 0.146 second response time
- 14:44: cdanis destroys php7 opcache on mw1271, mw1256
- 14:44: Icinga reports RECOVERY - PHP7 rendering on mw1256 is OK: HTTP OK: HTTP/1.1 200 OK - 75405 bytes in 1.031 second response time

## Conclusions

Is a partial opcache clear more risky wrt: corruption than clearing the entire opcache?

### What went well?

- automated alerts on PHP7 rendering and logstash fatals detected the incident
- elukey noticed the above quickly

### What went poorly?


- no PHP7 rendering alert for the wikidata-specific code that was corrupted on mw1271

### Where did we get lucky?

- opcache corruption was not / has not been more widespread or more frequent
- the nonsense string 'UseKeyHe`der' was a good clue that something like this was to blame

## Links to relevant documentation



*Where is the documentation that someone responding to this alert should have (runbook, plus supporting docs). If that documentation does not exist, there should be an action item to create it.*

- Previous instance of opcache corruption of interned strings: [T221347](#) 

## Actionables

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

**NOTE:** Please add the [#wikimedia-incident](#) Phabricator project to these follow-up tasks and move them to the "follow-up/actionable" column.

- set `opcache.validate_timestamps` to check opcache-vs-disk timestamps, instead of scap issuing `opcache_reset` commands  **Done**
- is scap sync-file with a partial opcache clear more dangerous than clearing the whole opcache?
- continue work on dbctl, so we need many fewer mediawiki deploys
- PyBal should check both HHVM and PHP7 backend URLs to determine host health [T222705](#) 

Category: [Incident documentation](#)

This page was last edited on 7 May 2019, at 21:01.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About](#)

[Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)

Wikitech

