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/20141118-EventLogging

< Incident documentation

Contents [hide]

Summary

Accidentally deployed EventLogging code (for which the database writer failed after some time) rendered EventLogging's writing of events to the database unreliably between 2014-11-18T00:59 -- 2014-11-13T01:21.

Timeline

2014-11-18T02:10

Reviewed but untested EventLogging code got deployed unintentionally.

2014-11-18T02:12

EventLogging's database writer failed.

2014-11-18T02:37

EventLogging's database writer recovered. Events got written to the database again.

2014-11-18T07:26

EventLogging's database writer failed.

2014-11-18T11:43

EventLogging's database writer recovered. Events got written to the database again.

2014-11-18T13:44

EventLogging's database writer failed.

2014-11-18T15:02

EventLogging's database writer recovered. Events got written to the database again.

2014-11-18T18:16

EventLogging's database writer failed.

2014-11-18T19:48

Ryan sent a notice 🗗 to the analytics list about EventLogging events do not showing up in the database.

2014-11-18T21:05

EventLogging's database writer recovered. Events got written to the database again.

2014-11-18T23:19

EventLogging's database writer failed.

2014-11-18T23:55

QChris saw Ryan's message to the mailing list.

2014-11-19T00:07

EventLogging's database writer got restarted manually. Events got written to the database again.

2014-11-19T00:36

Ori deployed a working EventLogging version.

Conclusions

- EventLogging deployments are anonymous. But as (regardless of who deployed) only an existing, reviewed commit from gerrit and not rogue code got deployed, it seems the deployment happened accidentally by trusted people, and was not ill-intentioned. Also, this is the first time it happened for EventLogging. So no need to take measures against it.

- Monitoring of EventLogging processes is not sufficient. We need monitoring that checks if events actually end up in the database. That's on our to-do list anyways.

Actionables

- Status: ▌ **Done** Make sure that log files exist for the affected period, so we can backfill.
- Status: ▌ **Done** Backfill the data.
- Status: ▌ **Unresolved** Implement monitoring that makes sure that the events end up in the database.

Categories: Incident documentation │ EventLogging/Incident documentation