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

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/20190715-logstash

< Incident documentation

document status: final

Contents [hide]

Summary

The Elasticsearch instance backing logstash began rejecting messages from cpjobqueue and changeprop due to a type error on the 'error' key. Mediawiki logs were unable to be consumed because logstash continually retried these failing logs and had no additional capacity to consume off the queue. Backlog of messages in Kafka increased steadily to over 18M messages until a configuration change to drop all messages from changeprop and cpjobqueue was implemented.

Type collisions in Elasticsearch have been a long-standing issue but haven't bit Mediawiki logging this badly before [*citation needed*]

Impact

Once the issue was discovered, deployments were blocked (some deployments proceeded while the issue was ongoing, but before discovery). For the interval we had drastically limited visibility into Mediawiki errors. Visibility of most logs delayed many hours. Believed no logs lost forever, aside from those deliberately dropped (changeprop and cpjobqueue).

Detection

Human detection after six hours. No automated alerts.

Timeline

All times in UTC.

- Before 12:27 **OUTAGE BEGINS**
- ~13:00 Determine that logs are backing up in kafka but are not being consumed by logstash
- ~19:20 Decision to drop logs from cpjobqueue and changeprop.
- 19:37 Logstash bad-logs-drop configuration enabled
- ~00:00 (Jul 16) Logstash has consumed the queue. **OUTAGE ENDS**

Graphs

- Kafka consumer lag: a measure of processing backlog
- Logstash metrics

Some log excerpts

```
[2019-07-15T17:16:42,892][WARN ][logstash.outputs.elasticsearch] Could not index
event to Elasticsearch. {:status=>400, :action=>["index", {:_id=>nil,
:_index=>"logstash-2019.07.15", :_type=>"cpjobqueue", :_routing=>nil}, 2019-07-
15T17:16:39.237Z scb1004 Error during deduplication], :response=>{"index"=>
{"_index"=>"logstash-2019.07.15", "_type"=>"cpjobqueue", "_id"=>"AWv2ouYQ-
pVO1Pagju_E", "status"=>400, "error"=>{"type"=>"illegal_argument_exception",
"reason"=>"[error] is defined as an object in mapping [cpjobqueue] but this name is
already used for a field in other types"}}}}
```

Conclusions

What went well?

- Not all too much

What went poorly?

- The issue was found by two volunteer deployers, both of whom reported it on IRC in #wikimedia-operations. However, the severity of this issue was not understood and thus for 6 hours nothing was done about it.
- The issue was not flagged by automated monitoring in any way.

Where did we get lucky?

- Several deployments took place during the Logstash outage. In that time, a MediaWiki configuration change or software deployment could have caused a full or partial Wikipedia outage that could take quite long to find and mitigate given absence of log analysis (other systems likely would have detected a broad outage, but we'd not have information in Logstash to correlate the issue, find how common or serious it is, who is affected, stacktraces, timestamps, etc).
  - During SWAT deployments, mwdebug servers are used to manually verify changes (with a dedicated Logstash dashboard). This dashboard is empty by default due to most requests not causing errors and the debug servers not receiving other traffic. When this remains empty whilst verifying a change prior to deployment, that means it is OK. Whereas in this case that would have been a false positive.
  - Scap normally detects and aborts the deployment based on canary servers and Logstash checks. These checks return "All OK" because Logstash was empty.

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.

Actionables

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

- Alert in icinga on an expected rate of incoming Mediawiki log messages, likely by executing a query against kibana for the past N minutes -- this is currently handled by logging rate checks
- Something to ensure we don't have conflicting-schema time bombs waiting to happen to us again bug T150106
- Alert on logging kafka queue depth. bug T228145
- Enable the Kafka dead letter queue for logstash -- Something like this would not have helped us in this case (see bug T150106). bug T223483 is related as to why the lag increased, but the proper solution is bug T234565

Category: Incident documentation