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/20150615-Elasticsearch

< Incident documentation

Elasticsearch outage: bug T102463

Summary

Elasticsearch service (on elastic*.eqiad.wmnet nodes)-ES for short-, backing the search functionality on all wikis was partially down for 6 hours, and fully unavailable to the users for 3 hours, starting at 9:26 UTC.

Timeline

Server admin log entries during this time

- Near 7 UTC, some ES hosts start to fail, as noted by 5xx spikes on mediawiki:

```
[06:59:07] <ori>  elastic is unhappy ... 5xx spike appears related ... mw
exception log has things like:
[06:59:55] <ori>  MWException from line 184 of /srv/mediawiki/php-
1.26wmf9/extensions/GeoData/api/ApiQueryGeoSearchElastic.php:
Elastica\Exception\PartialShardFailureException at /srv/mediawiki/php-
1.26wmf9/vendor/ruflin/elastica/lib/Elastica/Transport/Http.php
[07:01:18] <ori>  more specifically, having a shard fail prompts the
CirrusSearch extension to try and report a fail status, which then prompts
"Fatal Error: Class undefined: CirrusSearch\Status"
```

- Icinga fails to acknowledge some of that failures for a long time, as appointed by ori and later by joe:

```
[07:12:04] <_joe_>  note that on the "failing"
[07:12:13] <_joe_>  hosts, ES reports the cluster "green"
[07:12:23] <_joe_>  even if it takes it 1 minute to answer
```

- Ori hot-patches Mediawiki so that frontends do not throw an exception if the searches fail bug T102454:

```
07:21 logmsgbot: ori Synchronized php-
1.26wmf9/extensions/CirrusSearch/includes/Util.php: I504dac0c3: Add missing 'use
\Status;' to includes/Util.php (duration: 00m 13s)
[07:26:18] <ori>  so since I deployed my patch, 5xxs went down, because the
search failures are no longer causing fatals
[07:26:22] <ori>  but searches are still failing
```

- But that was a (bad) consequence, not the underlying cause, which was ES fully failing (or at least many nodes, including the masters).

```
[07:06:51] <ori>  when trying to search: "An error has occurred while searching:
Search is currently too busy. Please try again later. "
```

- Single node restarts do not work, as they cannot join back to the cluster. Logs show the master as

unresponsive:

```
[07:26:42] <icinga-wm>  PROBLEM - ElasticSearch health check for shards on
elastic1013 is CRITICAL - elasticsearch http://X:9200/_cluster/health error
while fetching: HTTPConnectionPool(host=X, port=9200): Read timed out. (read
timeout=4)
Jun 15 09:07:23 <moritzm>       after the restart it's still throwing the
MasterNotDiscoveredException
```

- We decide to soft-disable search functionality, as searches are blocked and their queue full:

```
09:26 logmsgbot: oblivian Synchronized wmf-config/PoolCounterSettings-
common.php: temporarily throttle down cirrussearch (duration: 00m 13s)
```

- There has been garbage collection issues on many nodes:

```
Jun 15 10:25:59 <moritzm>       some hosts like 1003 are not affected, gc never
took longer than 2 seconds: https://phabricator.wikimedia.org/P782
```

- We finally locate an EL expert, which confirms the issues and solves them by killing all offending nodes:

```
Jun 15 10:23:03 <manybubbles>   _joe_: I'm going to restart the master - it
seems hosed up. not sure how yet
```

- That doesn't really fix it. Master keeps getting hosed. Not sure why. I restart other master. Cluster become super unhappy. I issue a full cluster restart.
- Now the cluster is at least in a recovering state, unlike before. Recovery speed is increased but it takes ~2-3 hours to resynchronize the cluster with at least 2 replicas for the enwiki of every shard.
- It is put back up (throttled) when all shards have at least one replica, and fully up a bit later

```
13:11 logmsgbot: demon Synchronized wmf-config/PoolCounterSettings-common.php:
all the search (duration: 00m 12s)
12:52 logmsgbot: demon Synchronized wmf-config/PoolCounterSettings-common.php:
partially turn search back on (duration: 00m 13s)
```

- at 15:00, all ES hosts are healty (Green)

```
15:00 _joe_: ES is green
```

The next morning (EU time) it happens _again_. Pretty much rinse and repeat but everything is faster. We find out the root cause and take some (soft) corrective action to make sure it doesn't happen again. We're not fully protected yet, but we'll schedule work to fix it.

Thanks to all people involved: Giuseppe, Nik, mobrovac, Chad, moritzm, Mark, Max and others

Conclusions

- ES monitoring does not reflect properly the state of the cluster: it does not warn in yellow state, and general health monitoring was not enough to detect this particular case
- ES topology could be improved, as suggested by several people: things like master nodes not being data nodes, and maybe decoupling more wiki searches?
- Difficulty of testing ES java configurations, such as gc settings
- Ganglia tie-in for ES stats is error-prone and gets in the way during an outage
- Root cause is https://phabricator.wikimedia.org/T102589 - its not visible to those who haven't signed the NDA.
- There are other issues - mostly the elasticsearch doesn't recover properly even after we bounce the nodes that crash from the query. I'm tracking that here: https://phabricator.wikimedia.org/T102594 . Its also N
  - This issue also came up in 20141027-CirrusSearch which was this issue but with a different query.

Actionables

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

- Status: **Done** Improve MW reliability on search failures (bug T102454)
- Status: **Done** Search shouldn't display misleading "No results found" if there is an error in the search backend (bug T102464)
- Status: **Unresolved** Improve ES icinga monitoring. Icinga should probably detect "gc death spirals" occuring on ES nodes (e.g. by detecting GC runs running than x seconds) (https://phabricator.wikimedia.org/P782 list the ones that occured during the outage)
- Status: **Unresolved** Improve ES reliability with an architecture change? Related commit: https://gerrit.wikimedia.org/r/218421
- Status: **Unresolved** Improve documentation/operations involvement with EL?
- Status: **Unresolved** ES metrics should be pushed from master instead of polling with Ganglia. Use statsd plugin. (bug T90889)

Category: Incident documentation