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/20131118-CirrusSearch

< Incident documentation

Contents [hide]

## Summary

From 16:00 UTC to 20:00 UTC CirrusSearch was broken for several periods tens of minutes in length. Wikis with Cirrus as primary experienced search and update problems during that time. Wikis with Cirrus as secondary experienced update problems during that time.

## Timeline

*See also: Server admin log entries during this time.*

A few weeks ago we disabled redundancy by default in CirrusSearch because it was causing issues with puppet. We enabled it via configuration in production. This configuration didn't take.

Late last week Elasticsearch 0.90.7 was released and I asked Andrew Otto to add it to apt so I could test it. The testing went perfectly in my development environment and beta including a rolling upgrade during the testing. This testing working is suspicious given that we didn't have any redundancy in either dev or beta at the time. I'm really unsure how this upgrade didn't catch that. As this came late in the week I was not able to secure a window to upgrade production that week.

Chris Johnson installed the Elasticsearch machines over last weekend. When he set them up in puppet they automatically joined the Elasticsearch cluster and data was migrated to them like any other node. Note now that we are running both Elasticsearch 0.90.4 and Elasticsearch 0.90.7 in the cluster. This is supposed to work, at least well enough for your to perform a rolling upgrade. When I discovered this I figured we should just finish decommissioning testsearch100[1-3] so we can get them off the cluster.

Around 16:00 UTC to properly set up the partition on the Elasticsearch machines Andrew Otto took down elastic1001 and blasted the data directory and mounted the appropriate disk there. I had assured him this would be ok as Elasticsearch had two other copies of this data and would replicate it back out. I also assured him that keeping the data around wouldn't really help with anything because Elasticsearch tends to not take into account the old shard state when assigning shards to machines that join back up. It is a long standing complaint folks have with Elasticsearch.

At 16:15 UTC icinga warned that Elasticsearch was in "red" state, meaning there were unassigned master shards. I started investigating and I discovered that we'd turned off all the redundancy. The records that Andrew had deleted were gone. In hindsight it is at this point I should have switch back to lsearchd. We'd lost 1/15 of the search index in such a way rebuilding it will take a few hours at minimum.

Rather than switch to lsearchd I figured I could stop the bleeding by making sure all the nodes were assigned. Some of the search index would be missing but at least things would be moving again. After a few minutes of doing this manually I wrote a script to assign all the unassigned shards. Elaticsearch doesn't have a big "I've broken it. I know I'm going to lose data, just bring these back online now." button and doing each one by hand took too long. The script lives here: https://wikitech.wikimedia.org/wiki/Search/New#Stuck_in_red may it never be useful again.

At 17:01 UTC I got all the shards assigned. They were empty but we were going to perform a rebuild anyway. All elasticsearch wikis were limping along now.

At 17:40 UTC Chad fixed the redundancy configuration error and I kicked off a process to bring new replicas on line. I working on something that looked pertinent but ultimately didn't matter.

At 18:09 UTC icinga warned again of Elasticsearch having unassigned master shards. I spent a while digging into this and discovered that there was some kind of error moving replicas from Elasticsearch 0.90.7 to 0.90.4. It looked like a bug caused by them upgrading Lucene. I haven't dug into it yet. This problem caused nodes to stay unassigned.

Around 18:40 UTC I configured Elasticsearch to remove any data on those nodes and at 18:50 UTC Elasticsearch recovered again, having assigned all nodes replicas.

Around 19:00 UTC Andrew Otto and I went back to performing the rolling restarts to fix the partition. We used a command that moves shards off of the node before we shut it down. This seemed to work fine for elastic1002.

testsearch100X now known evil and no longer holding any data I amended a puppet change for properly configuring elastic10XX to decommission testsearch100X as well and asked Andrew to review it again. At the time I believe that testsearch1001 being the cluster master was part of the cause of the replica assignment issue.

He merged it at 19:18 UTC. I started seeing fatals around 19:25. Paravoid pointed me to a icinga CRITICAL for the lvs pool. It turns out I never added the new nodes to LVS. At this point testsearch1001 and testsearch1002 had been decommissioned. I thought Chris had added them when he set up the new nodes this weekend. I was convinced adding the lvs::realserver class was enough to do this and that he had added the class when he set up the new machines. Not only had he not added the class, that class doesn't set up lvs! Mutante and paravoid synced a corrected lvs config at 19:31 UTC.

At 19:35 UTC it was obvious this fix wasn't working. We were still getting fatals. For a few minutes I believed they were caused by requests being stuck in the queue. Around 19:45 UTC I realized that we had unassigned shards again. I started the process of reassigning them again, with the script. At 19:52 UTC the fix finished and we began talking about removing Cirrus from all wiki until we could figure out what happened. At 19:56 UTC Chad did just that.

I started writing this at 19:02 UTC.

## Conclusions

Original cause:

- Configuring 0 replicas. This was a time bomb waiting for the first server failure or non-super careful upgrade.
  - Issue fixed.
  - We should add an icinga alert for this case (Bugzilla:57210)

Cause of delay in recovery:

- Half completed upgrade combined with so far un-traced error in Elasticsearch
  - I need to schedule the entire upgrade before importing anything into apt.
  - Can we have a separate apt for labs?

Cause of second outage:

- Decommissioned servers without removing them from LVS.
  - I need to learn more about LVS so I don't do this again. So should Andrew Otto.
  - It'd be nice if lvs::realserver's documentation also had something like "this does not add this machine to an LVS pool. Please read XXXX for more information." or something.

Cause of third outage:

- Unknown shard failures.
  - Investigating.

## Actionables

- Someone (Chad?/Andrew Otto?/Me?) needs to come up with a set of conditions we can use to decide when to roll back to lsearchd earlier and somehow learn from my hindsight.
- Before we turn off lsearchd for good we need some equivalent fallback for Cirrus. Is this running Elasticsearch at a second datacenter? We'd talked about doing that at some point.
- The icinga plugin for Elasticsearch needs some love. bugzilla:57210
  - It'd be nice if it managed the Elasticsearch cluster as a whole rather than per server so it doesn't spit out warning for _each_ server.
  - *From Antoine's reply*

    - There is a plugin to monitor clusters. Use case, doc, examples at: http://docs.icinga.org/latest/en/clusters.html and https://www.nagios-plugins.org/doc/man/check_cluster.html
    - The idea is to create a service that is based on the result of other services.
- It'd be nice if it didn't spit out a huge blob of json on failure.
- It'd be nice if it detected a split brain. We didn't have one but I was scared we might have for a while and it'd be nice to be able to just look at icinga.

## Bugs reported from this outage

- bugzilla:57210 - CirrusSearch: Improve elasticsearch monitoring
- bugzilla:57222 - CirrusSearch: Figure out why shards needed reassignment around 19:35 UTC Nov 18th
- bugzilla:57068 - CirrusSearch should skip the pool counter in maintenance scripts
- bugzilla:57215 - CirrusSearch: Fail updates quickly if Elasticsearch is down or otherwise broken
- bugzilla:56798 - CirrusSearch shouldn't use SQL to figure out page weight
- bugzilla:57221 - CirrusSearch: make sure to serve searches even if Elasticsearch is running without all shards

Category: Incident documentation