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/20190425-prometheus

< Incident documentation

Summary

Both prometheus@ops servers in eqiad simultaneously began allocating RAM at a prodigious pace until they were OOM-killed. The resulting monitoring artifacts created caused several SREs to believe some sort of network outage / lost traffic happened at eqiad. Prometheus returned to seemingly-normal operation a few minutes after being restarted by systemd.

It is unclear why the global prometheuses seemed to record bad data for the interval after restart.

The stack trace dumped by one of the prometheuses seems to suggest that enormous queries are to blame for the OOM.

As it turns out, loading the Kafka dashboard with 90 days of history will almost OOM the Prometheuses by itself -- refreshing the dashboard once with the first round of queries still in-flight will certainly do it.

Impact

Some SREs lost some time, some time intervals have inaccurate metrics

Detection

Pseudo-automated: the "traffic drop" Icinga alert fired, and the monitoring artifacts generated looked real enough that SREs were led astray for 30-45 minutes.

Graphs of said artifacts:

- https://grafana.wikimedia.org/d/000000479/frontend-traffic?orgId=1&from=1556224608805&to=1556225676342
- https://grafana.wikimedia.org/d/000000180/varnish-http-requests?from=1556224608805&to=1556225676342&orgId=1
- https://grafana.wikimedia.org/d/000000341/dns?orgId=1&refresh=1m&from=1556224608805&to=1556225676342

Timeline

All times in UTC.

- 19:00:54: final logging message from prometheus1003 prometheus@ops
- 20:33:58: first requests seen on `grafana1001` for the Kafka dashboard with 90 days of history
- 20:34: CPU utilization sharply increases and RAM consumption begins increasing at ~11GB/minute on both prometheus1003 and prometheus1004
- 20:41:42: OOM-killer on prometheus1003 kills prometheus@ops process
- 20:41:49: prometheus1004 prometheus@ops logs "fatal error: runtime: out of memory" and dumps a stack

trace

- 20:41:54: OOM-killer on prometheus1004 kills prometheus@ops process
- 20:44:44: post-restart prometheus1003 reports "Server is ready to receive web requests"
- 20:45:04: post-restart prometheus1004 reports "Server is ready to receive web requests"
- 20:45: first sign of traffic drop monitoring artifacts on global prometheus instances
- 20:48: "detection" by Icinga: `PROBLEM - Varnish traffic drop between 30min ago and now at eqiad on icinga1001 is CRITICAL: 39.71 le 60`
- 20:50: traffic metrics on global prometheus instances returned to approx. normal

Graphs:

- prometheus1003
- prometheus1004

Apache logs of long-running queries near the time of the incident:

- prometheus1003
- prometheus1004
- grafana1001

Conclusions

What weaknesses did we learn about and how can we address them?

The following sub-sections should have a couple brief bullet points each.

What went well?

- proximate cause was easy to find in the Apache logs on prometheus & grafana machines

What went poorly?

- no good dashboard for meta-metrics about Prometheus internals (there is a dashboard but it is not good)
- the default limits in place in Prometheus don't seem to be good enough at our scale?

Where did we get lucky?

- while prometheus ate up a ton of RAM and OOMed, it did so only once, not repeatedly

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

- Create a reasonable dashboard for prometheus internal operations T222102
- Roll out prometheus query limits (on CPU time, or RSS, or max tsdb points?) T222105 ✓ **Done**
  - After the above is completed: figure out what's so Prometheus-intensive about the Kafka dashboard and fix it T222112 ✓ **Done**
- Explicit IRC alerting for prometheus OOMs and other restarts, even if just to know to be wary of monitoring artifacts? T222108 ✓ **Done**
- Upgrade to prometheus 2.9.2 T222113

Category: Incident documentation