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/20170322-AuthDNS

< Incident documentation

Authoritative DNS was fully unavailable on 2017-03-22 from 13:34:43 to 13:37:26 (2m43s). It was at least partially degraded (< 3 servers responding) during the broader surrounding window of 13:30:36 to 13:38:55 (8m19s). As AuthDNS is an extremely deep layer of our infrastructure upon which almost everything else is a dependency, this also caused various secondary fallout for both internal and public services. There was a small spike of public-facing 5xx HTTP responses from about 13:35 -> 13:40 from these secondary fallouts. Overall incoming request rate doesn't seem to have been significantly impacted in the same graphs.

Contents [hide]

Summary

An authdns configuration change in the puppet repo was merged which contained a configuration error (which jenkins can't validate to prevent). After the initial puppet-merge of the bad change, the existing/underlying configuration of puppet and the systems themselves conspired to attempt automated restarts of all authdns servers with the new configurations over a ~4 minute window, causing the running daemons to stop and stay down as each was hit by the puppet agent runs.

Timeline

All times are UTC throughout.

- Graphs of 13:00 -> 14:00:
  - Authdns request stats: https://grafana.wikimedia.org/dashboard/db/dns?from=1490187600000&to=1490191200000
  - Public overall request rate and 5xx rates: https://grafana.wikimedia.org/dashboard/db/varnish-aggregate-client-status-codes?from=1490187600000&to=1490191200000&var-site=All&var-cache_type=All&var-status_type=5
- 13:19 BBlack self-merges bad change (https://gerrit.wikimedia.org/r/#/c/344140/). Faulty expectation that this was "safe" for the runtime servers even if the change had issues, because validation would happen either automatically or manually, before DNS daemon would get restarted to pick up the change.
- 13:30 ns3 stops in syslog: `13:30:36 eeden systemd[1]: Stopping gdnsd...`
- 13:32 ns3 alert: `< icinga-wm> PROBLEM - Auth DNS on eeden is CRITICAL: CRITICAL - Plugin timed out while executing system call`
- 13:33 Opsen responding on IRC, begin investigating -> fixing
- 13:34 ns2 stops in syslog: `13:34:15 baham systemd[1]: Stopping gdnsd...`
- 13:34 ns1 stops in syslog: `13:34:43 radon systemd[1]: Stopping gdnsd...`
- ~13:35 Several internal services (e.g. recdns, restbase, mobileapps, etc) begin soft-alerting via icinga of problems as secondary fallout
- 13:36 ns2 alert: `< icinga-wm> PROBLEM - Auth DNS on baham is CRITICAL: CRITICAL - Plugin timed out while executing system call`
- 13:37 ns3 online in syslog: `13:37:26 eeden gdnsd[11495]: DNS listeners started`
- 13:37 ns1 alert: `< icinga-wm> PROBLEM - Auth DNS on radon is CRITICAL: CRITICAL - Plugin timed out while executing system call`
- 13:38 ns1 online in syslog: `13:38:21 radon gdnsd[27942]: DNS listeners started`
- 13:38 ns2 online in syslog: `13:38:55 baham gdnsd[16501]: DNS listeners started`

- ~13:40 Most internal services / secondary fallout has recovered by this point in icinga logs

Conclusions

This incident highlighted some historical weak-points that had gone unnoticed, and some newer work has further weakened and aggravated the situation:

- Our puppet agents, which apply configuration changes, run automatically once every 30 minutes on all hosts. The timing is statically randomized (as in, each host gets a certain consistent-over-time random number) across the entire fleet as a whole to spread puppetmaster load. As it happens, the random timing chosen for our 3-node AuthDNS cluster places all three of them running their agents within a 5-minute window. This leaves very little time to notice and correct any problem (even one caused by unrelated commits, e.g. a wmflib or base -class update) which might break all three authdns servers as the automated agent runs happen.
- Historically, we've never had CI for the AuthDNS configuration files managed directly by puppet. This was considered acceptable at some distant point in the past (and went largely un-reexamined later) for three reasons:
  1. Puppet was not configured to restart the DNS servers after pushing a config change (that was left to manual admin action)
  2. The service initscript and daemon both did late-stage validation of the configuration before attempting to stop/restart the running service, preventing actual runtime fallout even if a faulty change had been merged and manual admin update was attempted.
  3. There was only one very simple config file in question whose contents did not change frequently
- Nov 2014: A puppet change was merged that allowed puppet to restart the authdns service automatically when the configfile changed, removing protection (1) above. This was deemed acceptable because the other safety nets (2) and (3) above were still in play. The upside was that it avoided the state of affairs where someone might puppet-merge a broken config change and fail to take manual action shortly afterwards, leaving a failure-at-a-distance for someone to discover much later during the next attempt to change DNS configuration/data.
- ~Mid 2015: The Authdns servers were migrated from Ubuntu Trusty to Debian Jessie. One of the key changes here was the replacement of upstart with systemd. For reasons that would take a great deal of time and depth to explain, under systemd we lost the initscript/daemon guards (protection (2) above) against attempting potentially-outage-causing stops or restarts when a bad configuration file was in place. That loss is not a problem there is any easy solution for. However, we still had a validation step in place during authdns-update, which is the mechanism for almost all DNS changes (zonefile edits, geoip mapping changes, etc). The only saving grace left at this point on puppet-driven config changes is protection (3) above: that there's still only one small and infrequently-changing config file managed by puppet directly, so we're unlikely to trip over this hidden issue in practice.
- Recently: Work on DNS-based discovery tooling for multi-dc / dc-switching goals added multiple new and more-complex authdns configuration files managed by puppet. The stanzas are copypasted from the existing puppet-managed configfile, and thus also can trigger automated and outage-inducing restart failures.

All of the above conspired to create the observed situation: a seemingly-harmless puppet commit is merged, changing one of the puppet-managed authdns config files. The linter doesn't support checking its validity. Merging the change starts an implicit and unexpected countdown to destruction: 11 minutes later, the first authdns server automatically applies the bad change and outages itself attempting an unchecked restart. In less than 5 minutes, the other two follow suit, causing complete authdns outage in too short a timeframe for reasonable human prevention (at that point).

Actionables

- Status: 🟩 **Done** Fix the faulty commit that triggered the problem (https://gerrit.wikimedia.org/r/#/c/344148/)
- Status: 🟩 **Done** Remove puppet's ability to auto-restart AuthDNS daemons when it changes config files (https://gerrit.wikimedia.org/r/#/c/344152)
- Status: 🟥 **pending** Fix the cron-timing issue somehow Task T161145
- Status: 🟥 **pending** Refactor authdns CM/CI infrastructure Task T161148

Category: Incident documentation

This page was last edited on 22 March 2017, at 19:35.

details.