Closed Opened 1 year ago by **David Smith** 🌴

RCA for June 2nd GCP related GitLab.com incident

@ahanselka and @ahmadsherif - would you be okay owning this RCA? I'll add questions on my side too. cc @andrewn

Summary

Google had a major networking outage in their US East regions that affected their entire infrastructure, including GitLab.com

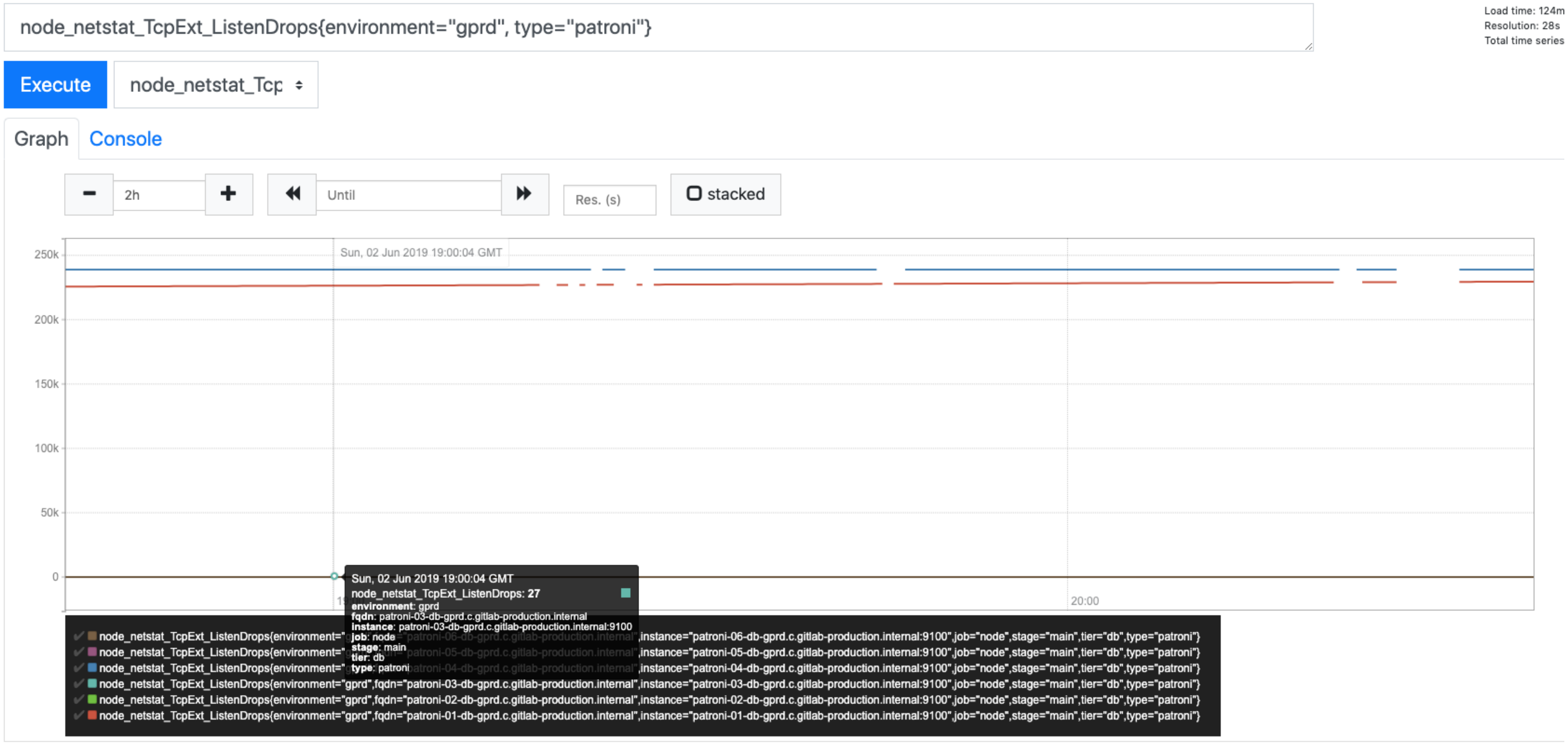
- Service(s) affected : All services for GitLab.com
- Team attribution : External
- Minutes downtime or degradation : 20 minutes of downtime, 190 total of degradation

https://dashboards.gitlab.net/d/ZUei7TkWz/platform-metrics?orgId=1&fullscreen&panelId=3&from=1559498453907&to=1559516462767

Impact & Metrics

Start with the following:

- All GitLab.com services were completely down for about 30 minutes as Postgres failed over and services had to be restarted/HUPed. The application had degradation for about 3.5 hours after that.
- All users of GitLab.com were affected by this incident.
- During the first 30 minutes, GitLab.com was entirely inaccessible to users. For the remainder of the incident there was an elevated rate of 5xx errors.
  - The main reason GitLab.com went completely down is due to Patroni cluster instability with failovers and the application being unable to follow the new primary.
  - The longer period of elevated rate of 5xx errors was related to the network instability.
- Many of our alerts were flapping because the monitoring server was unable to reach the servers in question.
- There was a lot of flapping between redis-cache primary and secondaries. This did not seem to have an affect on the availability of the application.

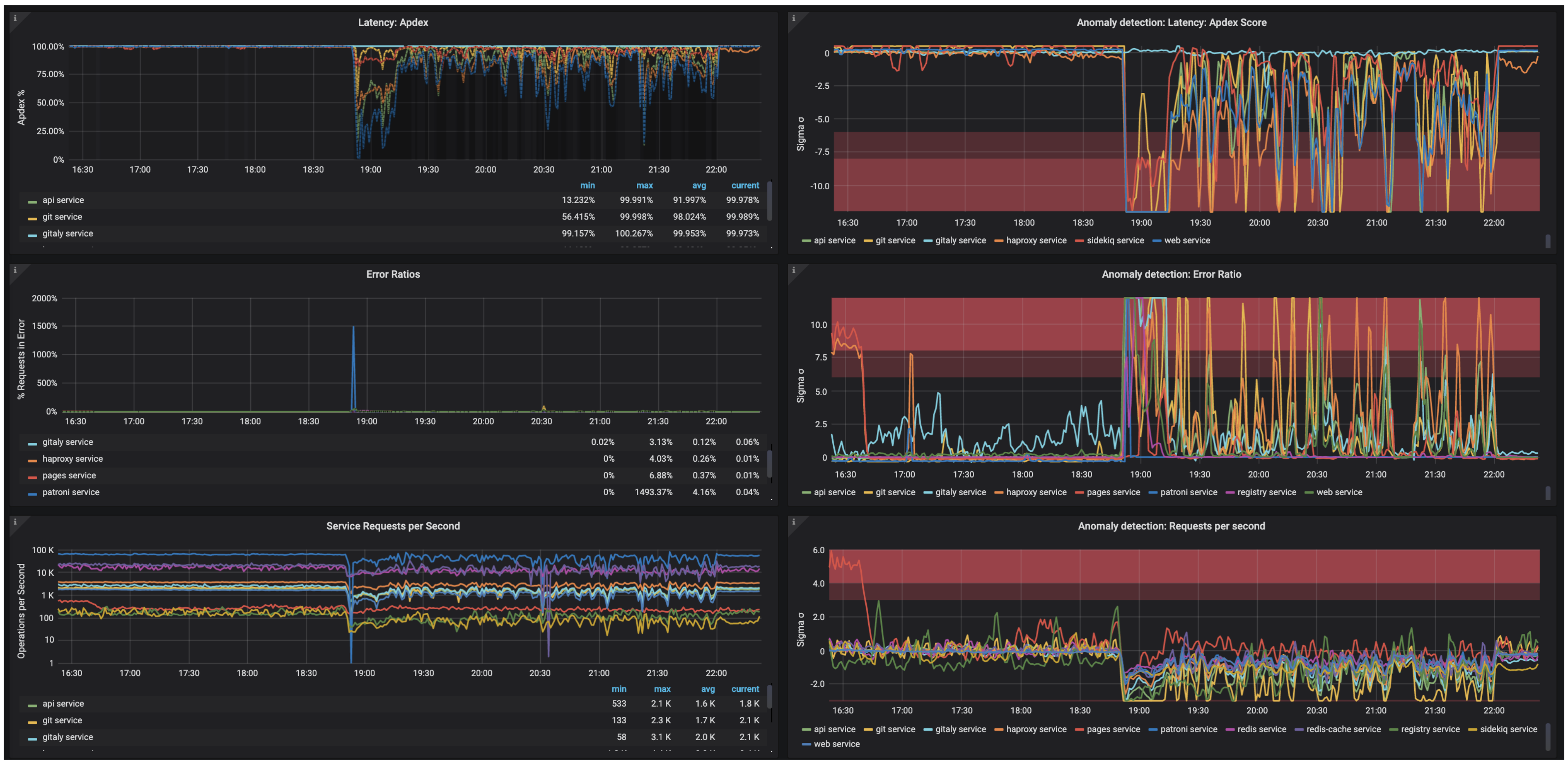


```
node_netstat_TcpExt_ListenDrops{environment="gprd", type="patroni"}
```

Load time: 124m
Resolution: 28s
Total time series

```
Sun, 02 Jun 2019 19:00:04 GMT
node_netstat_TcpExt_ListenDrops: 27
environment: gprd
fqdn: patroni-03-db-gprd.c.gitlab-production.internal
instance: patroni-03-db-gprd.c.gitlab-production.internal:9100
job: node
stage: main
tier: db
type: patroni
```

```
✓ node_netstat_TcpExt_ListenDrops{environment="  patroni-06-db-gprd.c.gitlab-production.internal",instance="patroni-06-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
✓ node_netstat_TcpExt_ListenDrops{environment="  patroni-05-db-gprd.c.gitlab-production.internal",instance="patroni-05-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
✓ node_netstat_TcpExt_ListenDrops{environment="  atroni-04-db-gprd.c.gitlab-production.internal",instance="patroni-04-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
✓ node_netstat_TcpExt_ListenDrops{environment="gprd",fqdn="patroni-03-db-gprd.c.gitlab-production.internal",instance="patroni-03-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
✓ node_netstat_TcpExt_ListenDrops{environment="gprd",fqdn="patroni-02-db-gprd.c.gitlab-production.internal",instance="patroni-02-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
✓ node_netstat_TcpExt_ListenDrops{environment="gprd",fqdn="patroni-01-db-gprd.c.gitlab-production.internal",instance="patroni-01-db-gprd.c.gitlab-production.internal:9100",job="node",stage="main",tier="db",type="patroni"}
```



	min	max	avg	current
api service	13.232%	99.991%	91.997%	99.978%
git service	56.415%	99.998%	98.024%	99.989%
gitaly service	99.157%	100.267%	99.953%	99.973%

gitaly service	0.02%	3.13%	0.12%	0.06%
haproxy service	0%	4.03%	0.26%	0.01%
pages service	0%	6.88%	0.37%	0.01%
patroni service	0%	1493.37%	4.16%	0.04%

	min	max	avg	current
api service	533	2.1 K	1.6 K	1.8 K
git service	133	2.3 K	1.7 K	2.1 K
gitaly service	58	3.1 K	2.0 K	2.1 K

Detection & Response

Start with the following:

- The incident was detected via PagerDuty alerts for GitLab.com being down.
- Alarming worked as expected, however there were so many alerts it was overwhelming and made it hard to quickly determine anything.
- Because of the deluge of alerts, some important relevant alerts such as the alert indicating Postgres failed over were lost in the noise.
- It took about 2-3 minutes after the beginning of the downtime to get alerted and begin response.
- It took about 30 minutes for us to recover GitLab.com from the Postgres failover, however the site remained unstable due to the provider outage.
- Our dashboards were partially broken, which was a known issue earlier in the week (production#849 (closed)), making it much more difficult to get started with the response.

Timeline

2019-06-02

- 18:50 UTC - Patroni failed over from -04 to -06
- 19:05 UTC - Most of our Grafana dashboards are inconsistently working because of thanos issues: production#849 (closed)
- 19:07 UTC - Pingdom returning errors
- 19:12 UTC - Diagnosis of Postgres failover
- 19:21 UTC - Services hup'd
- 19:26 UTC - GitLab.com operational. Pingdom reporting services as up.
- 19:41 UTC - Watching https://status.cloud.google.com/incident/compute/19003
- 19:59 UTC - also watching https://status.cloud.google.com/incident/cloud-networking/19009
- 20:39 UTC - Continuing to monitor google incidents
- 21:22 UTC - another failover from patroni-04 to -01
- 21:54 UTC - postgres failed back over to -04
- 22:00 UTC - Error rates returned to normal (that is, there were none)

2019-06-03

- 09:40 UTC - Restoring tuple statistics by running cluster-wide `ANALYZE`, see #5841 (comment 128321668) (done 10:10 UTC)

2019-06-06

- Google posted their RCA

Root Cause Analysis

GitLab.com went down for 30 minutes with instability over the course of 4 total hours.

1. Why? - The application could no longer reach the database.
2. Why? - Postgres failed over unexpectedly.
3. Why? - There was networking instability which caused the cluster to try to fail over multiple times.
4. Why? - GCP had a major networking outage in the east region where we are located.

What went well

Start with the following:

- We found out very quickly that there was a problem
- Multiple people jumped in to help diagnose and repair the issue
- Delegation of duties and expectations was clear and effective
  - i.e. "You go update the status page", "I will go restart sidekiq", etc.
- Patroni failover was successful. If we had still been on Repmgr it would have been true disaster.

What can be improved

Start with the following:

- We should better automate and tune our database failover process so that the application can gracefully handle failovers.
  - We could try to execute more failovers in staging and eventually production to be more confident that a similar incident in the future would not cause a complete outage as it did in this case.
- We should also automate such that we don't have to re-run `ANALYZE` on the tables to re-populate statistics.
- We can try to prune and curate our alerts such that there isn't a massive deluge of alerts that obscure the problem and make it hard to see other relevant alerts.
- While this did not directly affect production, we didn't notice that staging had fallen apart also as a result of this. #6854 (closed). It wasn't important to fix ASAP on a Sunday evening, but we should create a follow-up issue immediately when there is an issue like this so someone can follow up on Monday.

Corrective actions

Some of these issues are not created as a reaction specifically to this incident but are the correct actions.

- Automated failover testing (#5890)
- Clients still connect to old primary after failover (#5675 (closed))
- Graceful Patroni failovers (#5833 (closed))
- Production incident relating to Thanos problems (production#849 (closed))
- Use a virtual IP for failovers (#7059 (closed))

Guidelines

- Blameless RCA Guideline
- 5 whys

Edited 11 months ago by David Smith

Linked issues ? 1

Relates to

⊖ GitLab.com elevated error rates and site down for short period June 2, 2019

production#862

David Smith 🌴 @dawsmith changed milestone to %CI/CD & Enablement Team June 2019 1 year ago

David Smith 🌴 @dawsmith marked this issue as related to production#862 (closed) 1 year ago

David Smith 🌴 @dawsmith mentioned in issue production#862 (closed) 1 year ago

Alex Hanselka @ahanselka assigned to @ahmadsherif 1 year ago

Alex Hanselka @ahanselka · 1 year ago Owner

Yes, I'll start filling this out now.

🤖 GitLab Bot 🤖 @gitlab-bot · 1 year ago Maintainer

Hi @ahanselka,

This issue does not appear to have an issue weight set.

As a general guidelines use a weight of `1` for an access request issue or a simple configuration update. Use this as a multiplier for setting the weight. If you are unsure about what weight to set it is better to add a generous estimate and change it later. If the weight on this issue is `8` or larger then it might be a good idea to consider splitting this issue up into smaller pieces.

Thanks for your help! 🖤

You are welcome to help improve this comment.

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka changed weight to **4** 1 year ago

Alex Hanselka @ahanselka · 1 year ago Owner

> - 21:20 UTC - possibly another failover from patroni-04 to -01

@yguo can you confirm if there was actually another failover from 04 to 01?

yun guo @yguo · 1 year ago

yes it happened at 21:22:42 UTC

```
2019-06-02_21:22:42 patroni-01-db-gprd patroni[17041]:  server promoting
2019-06-02_21:22:42 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:22:42,277 INFO: cleared rewind state af
2019-06-02_21:22:43 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:22:43,291 INFO: Lock owner: patroni-01-
2019-06-02_21:22:43 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:22:43,389 INFO: no action.  i am the le
```

⌄ Collapse replies

Alex Hanselka @ahanselka · 1 year ago Owner

@yguo when did it fail back over to `patroni-04` ?

Please register or sign in to reply

yun guo @yguo · 1 year ago

but it failed over back to 04 at 21:54:55

```
2019-06-02_21:54:55 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:54:55,280 INFO: demoted self because DC
2019-06-02_21:54:55 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:54:55,281 WARNING: Loop time exceeded,
2019-06-02_21:54:59 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:54:59,118 INFO: establishing a new patr
2019-06-02_21:54:59 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:54:59,228 INFO: Lock owner: patroni-04-
2019-06-02_21:54:59 patroni-01-db-gprd patroni[17041]:  2019-06-02 21:54:59,229 INFO: does not have lock
```

⌄ Collapse replies

Alex Hanselka @ahanselka · 1 year ago Owner

Perfect, thank you!

Please register or sign in to reply

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka · 1 year ago Owner

I've filled this out fairly well. I'd love some feedback and/or updates from you @ahmadsherif and @andrewn .

cc/ @dawsmith

Edited by Alex Hanselka 1 year ago

David Smith 🌴 @dawsmith added 1 deleted label 1 year ago

David Smith 🌴 @dawsmith added workflow-infra In Progress scoped label 1 year ago

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka mentioned in issue #6854 (closed) 1 year ago

Andreas Brandl 🔴 @abrandl · 1 year ago Maintainer

Want to add that staging cluster was also severely broken after the network incident (only one replica was still intact), see #6854 (closed). Maybe it yields extra insights if we looked at the logs in staging more closely too (I want to but didn't have time just yet).

Collapse replies

Alex Hanselka @ahanselka · 1 year ago Owner

Thanks for finding and fixing this! I added this under things we can improve.

Please register or sign in to reply

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka · 1 year ago Owner

I added the Google RCA to the incident timeline.

David Smith 🌴 @dawsmith · 1 year ago Owner

@ahanselka and question for @gitlab-com/gl-infra. This incident and broad issue in GCP should make us ask a a question. Should we be running ops and/or dev in a different cloud provider (AWS)?

Also, I see ops mirroring runbooks and dev used to. Should we fix dev's mirror of runbooks?

cc @edjdev

Alex Hanselka @ahanselka · 1 year ago Owner

We already have an issue to put Ops in a different region of GCP. #6675. We haven't decided where to put it, but I think it should stay in GCP for sure. A majority of the GCP regions did NOT experience issues, including us-west1 where we are putting DR:

> Google Cloud instances in us-west1, and all European regions and Asian regions, did not experience regional network congestion.

Eric Johnson @edjdev · 1 year ago Owner

> but I think it should stay in GCP for sure

what's the reason?

What I was thinking is that having the infra team do it in AWS would create more knowledge about how our self-managed customers tend to do it. And so it would create more ability to help out if they hit scalability or even just config issues.

John Jarvis @jarv · 1 year ago Owner

> What I was thinking is that having the infra team do it in AWS would create more knowledge about how our self-managed customers tend to do it. And so it would create more ability to help out if they hit scalability or even just config issues.

I think there could possibly be benefit if we switch dev.gitlab.org to be a more "cloud native" with RDS for database, maybe EFS for shared filesystem, and EKS where we would use our own helm charts.

I'm not sure we are quite ready to do that though, and we also have slated dev.gitlab.org to be a "single VM omnibus installation" since as we move gitlab.com off omnibus we will need that validation.

The main reasons I have against it are biased I suppose because we aren't doing it well right now. We run multi-cloud right now because of our cloud provider migrations, this has left bread-crumbs of infra structure across:

- Digital ocean (chef server, for example)
- AWS (package cloud, some object storage buckets, plus a few more things)
- Azure (about, dev, etc)

While I understand the benefits, these are my reasons for wanting to move it to GCP

Security

- How we manage role based authentication, standardizing on things like IAP (in GCP) for authentication is slightly different or doesn't exist in AWS, this makes our management a bit more complicated.
- Having cross cloud network peering with ipsec is fine, but it adds a bit more overhead and adds complexity to the config
- Account access

Monitoring

- The lowest level monitoring we have is stackdriver metrics / cloudwatch metrics for GCP/AWS. Unlike the metrics we collect at a higher layer these are not as uniform and require specific prometheus exporters.

Runners

- This is specific to dev.gitlab.org, but runners are key and they will very likely be run in gcp, we should probably keep them close on the network.

Alex Hanselka @ahanselka changed the description 1 year ago

Casey Allen Shobe @cshobe · 1 year ago

If the application connects to a VIP which moves to the new master when a failover occurs, this should enable the application to failover immediately without being HUPed. This software adds the capabality for VIP management to Patroni: https://github.com/cybertec-postgresql/vip-manager

Alex Hanselka @ahanselka mentioned in issue #7059 (closed) 1 year ago

Alex Hanselka @ahanselka changed the description 1 year ago

Alex Hanselka @ahanselka · 1 year ago Owner

I created an issue for the VIP suggestion and added it to the corrective action section.

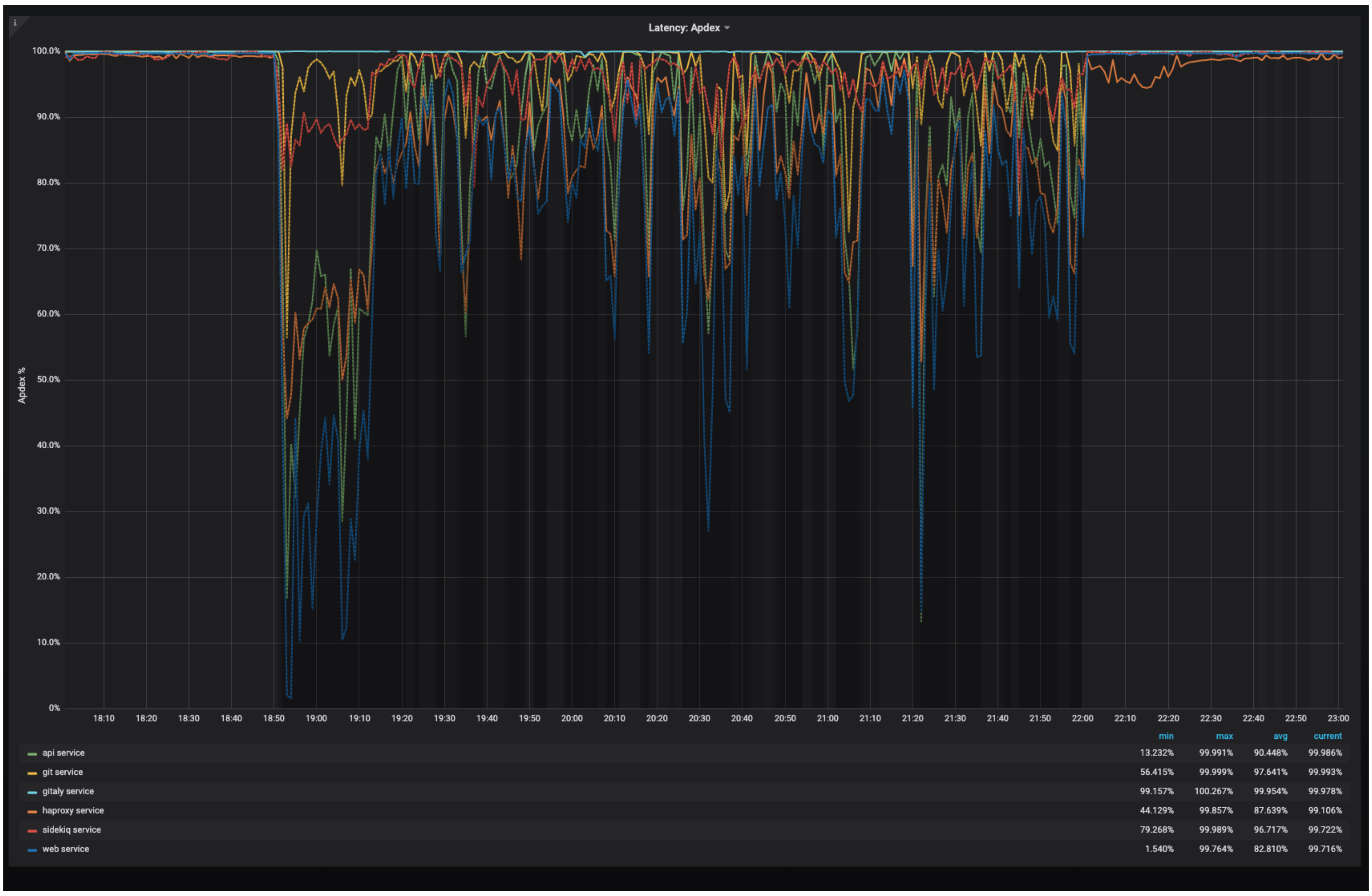
After discussing this in our meeting on Tuesday, I believe we can close this issue.

Alex Hanselka @ahanselka closed 1 year ago

ops-gitlab-net 💬 @ops-gitlab-net mentioned in issue #7040 (closed) 1 year ago

David Smith 🌴 @dawsmith changed the description 11 months ago

David Smith 🌴 @dawsmith · 11 months ago Owner



	min	max	avg	current
api service	13.232%	99.991%	90.448%	99.986%
git service	56.415%	99.999%	97.641%	99.993%
gitaly service	99.157%	100.267%	99.954%	99.978%
haproxy service	44.129%	99.857%	87.639%	99.106%
sidekiq service	79.268%	99.989%	96.717%	99.722%
web service	1.540%	99.764%	82.810%	99.716%

Anthony Sandoval @AnthonySandoval removed 1 deleted label 6 months ago

Anthony Sandoval @AnthonySandoval added team Reliability scoped label 6 months ago

Anthony Sandoval @AnthonySandoval reopened 3 months ago

Anthony Sandoval @AnthonySandoval added workflow-infra Done scoped label and automatically removed workflow-infra In Progress label 3 months ago

Anthony Sandoval @AnthonySandoval closed 3 months ago

ops-gitlab-net @ops-gitlab-net mentioned in issue #9472 (closed) 3 months ago

Please register or sign in to reply