

Closed

Outage due to half the fleet going dark

# Outage due to half the fleet going dark

We suffered an outage in which multiple hosts went dark and were not capable to come back on its own.

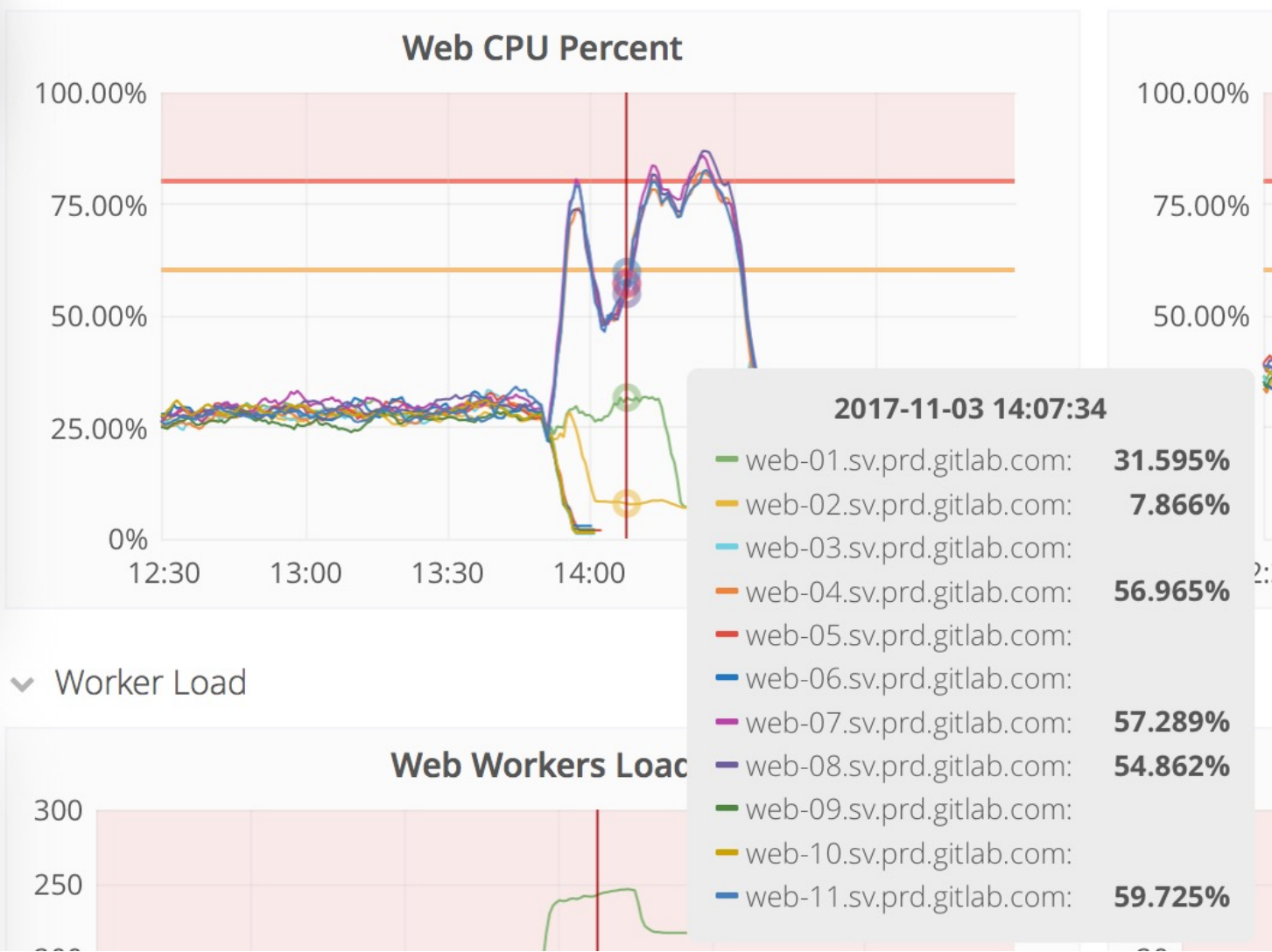
On top of this, the gitlab-pages processes locked in D state in web01 and web02 forcing us to bounce the whole hosts to recover the service.

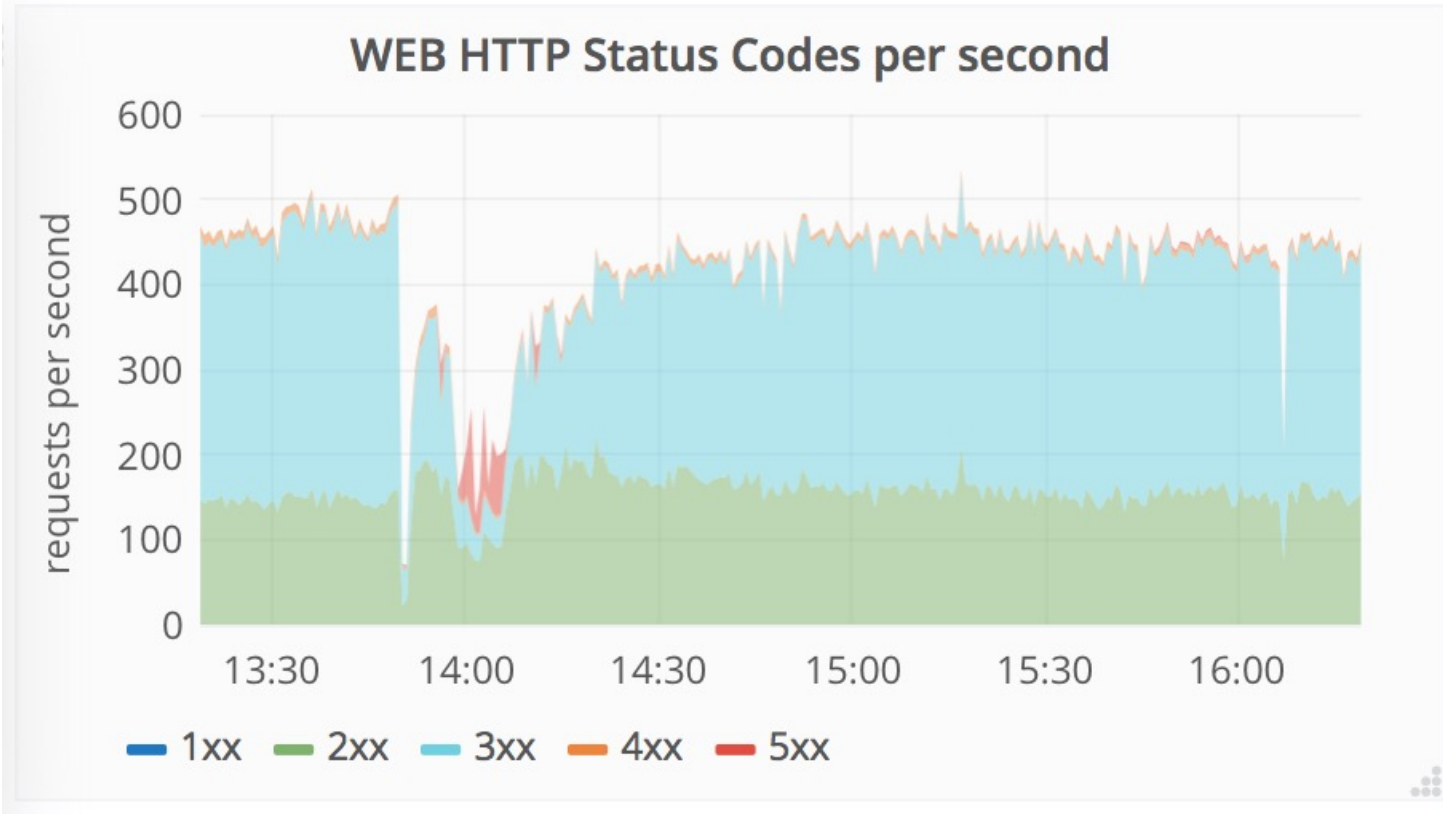
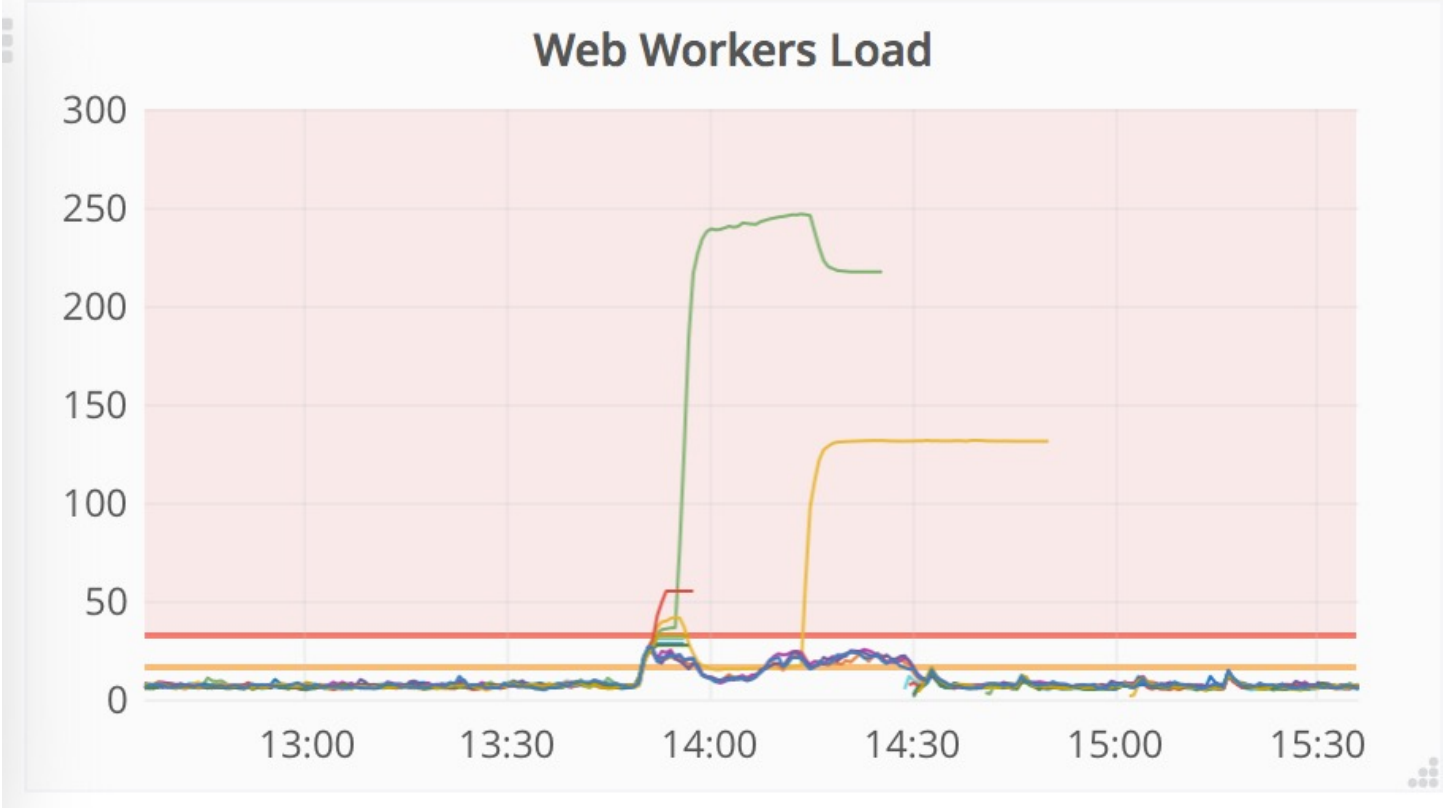
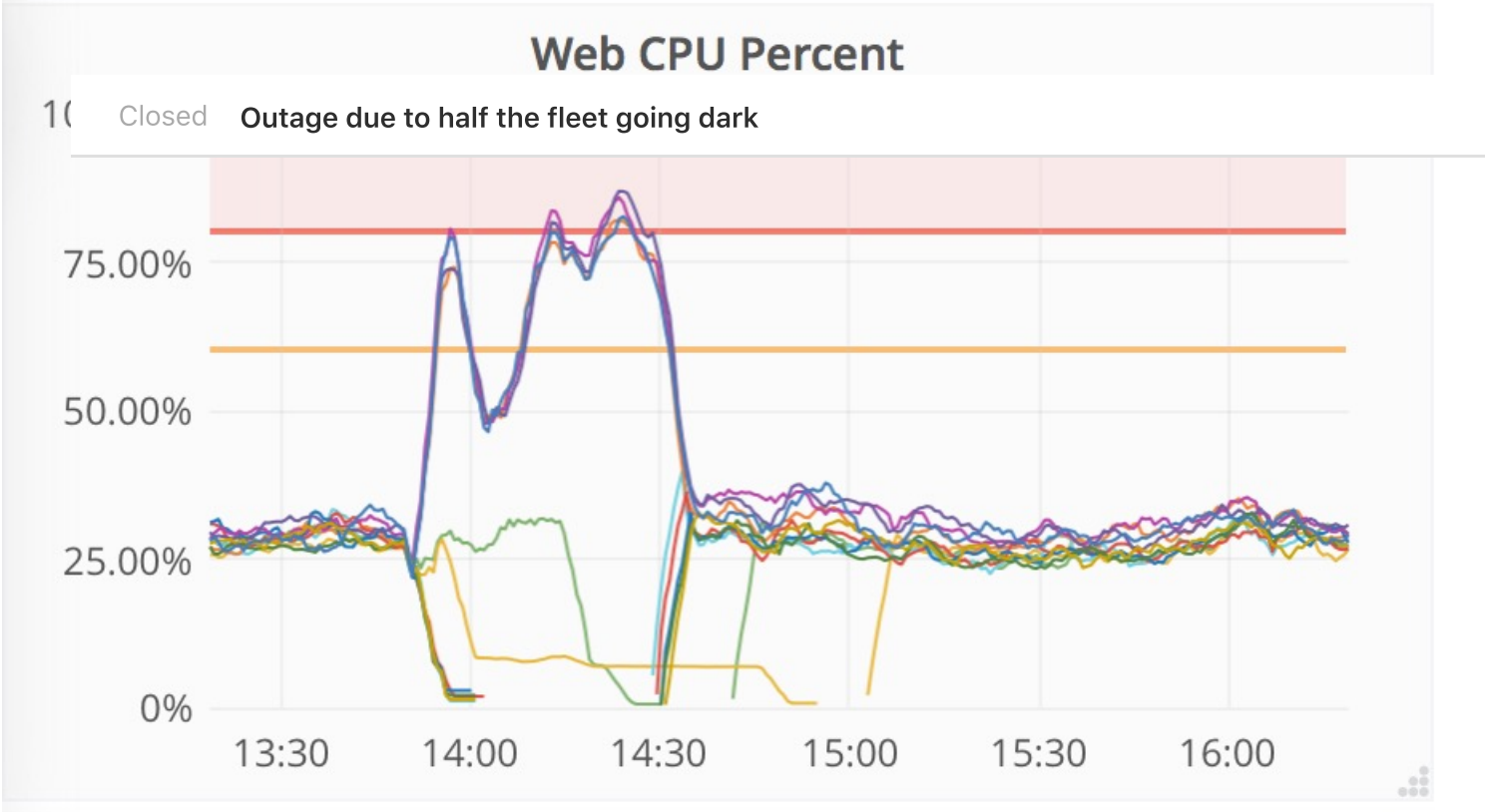
Everything points to a network partition that affected the pages NFS server, or only the web fleet, in any case, we suffered a service interruption with 6 minutes of downtime at GitLab.com, but almost 1h of downtime on the pages service.

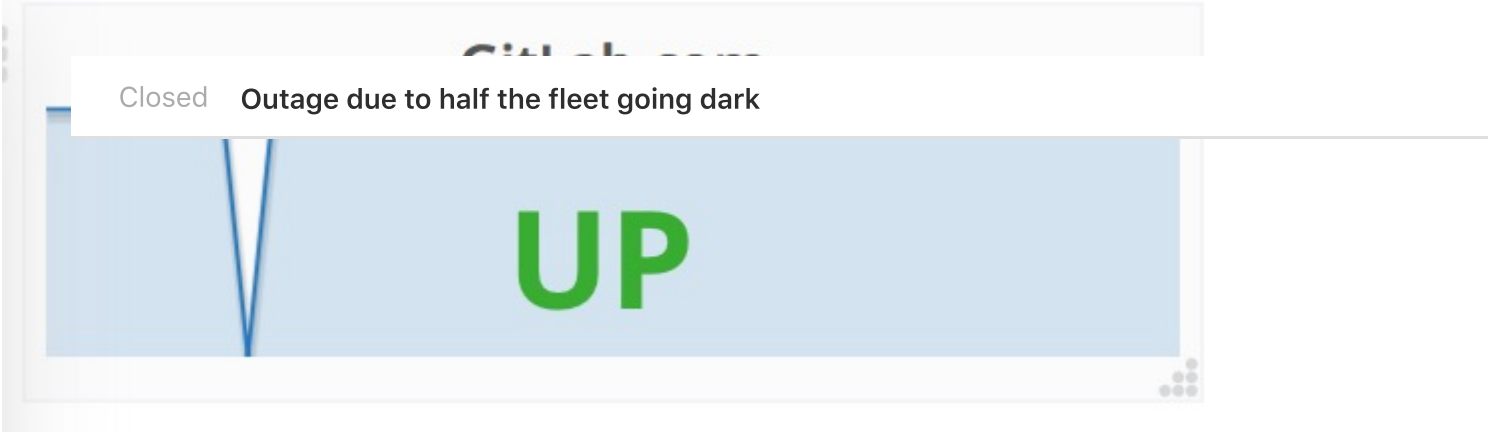
## Timeline in UTC

- 14:00 - We got paged by [@grzesiek](#) due to noticing errors 502 on GitLab.com. We immediately notice that some web workers had spikes on load (web01) and some were missing from monitoring.
- 14:08 - We ssh into the affected web worker and notice that almost all of the `gitlab-pages` processes are in D state, explaining the high load.
- 14:15 - We stopped `gitlab-pages` across the fleet to prevent these processes to impact the rest.
- 14:20 - After trying to recover web01 manually we decided to bounce it as it does not reduce load and uninterruptible processes can't be killed anyway.
- 14:23 - As web01 starts going dark, web02 load starts increasing.
- 14:31 - We decide to start bouncing all the rest of the web nodes that went dark before: web-03, web-05, web-06, web-09 & web-10
- 14:34 - The fleet recovers normal CPU usage patterns.
- 14:34 - We start seeing the CPU increase patterns that are coming from huge GC
- 15:06 - We recover the `gitlab-pages` service.

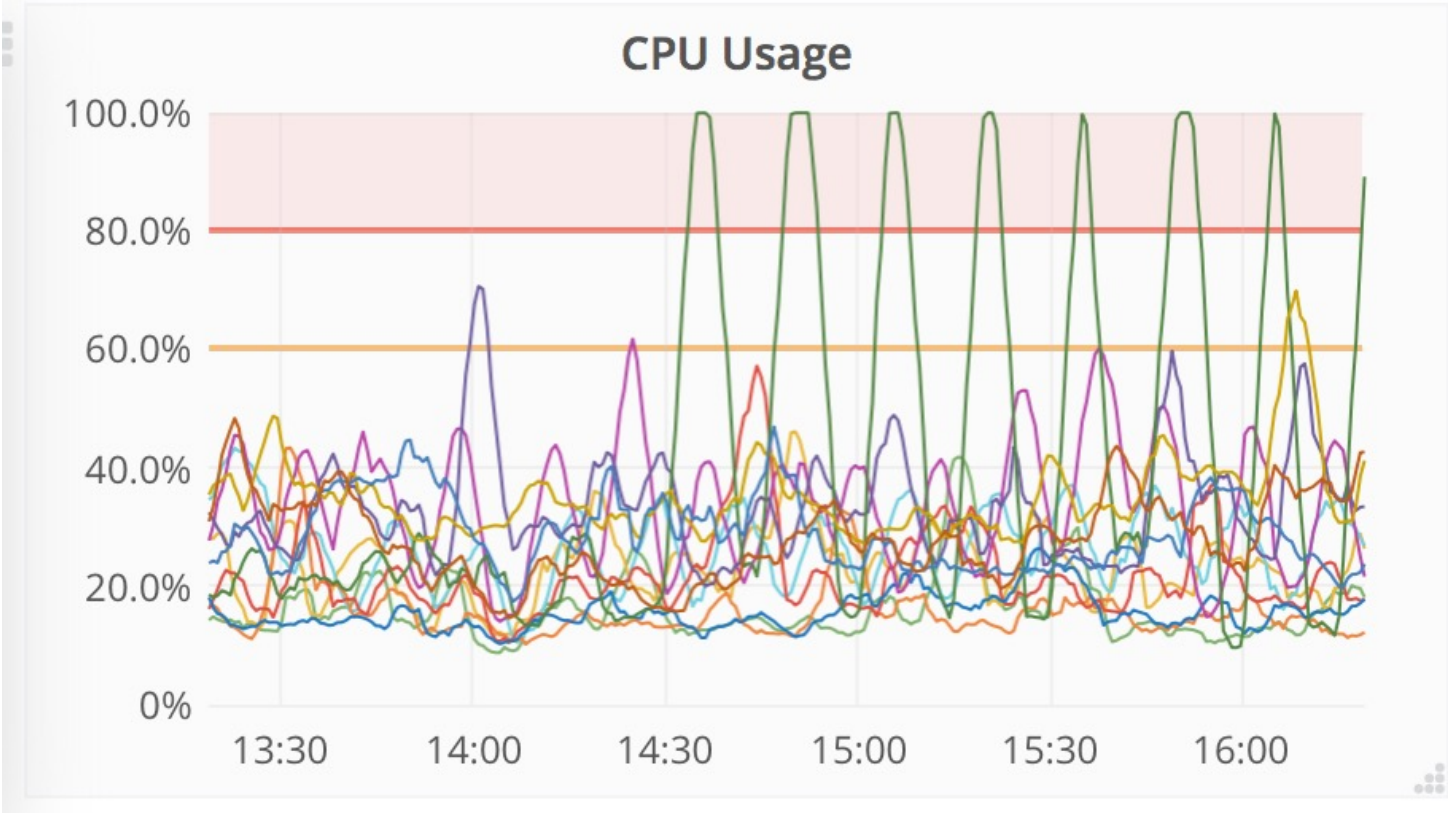
## Graphs



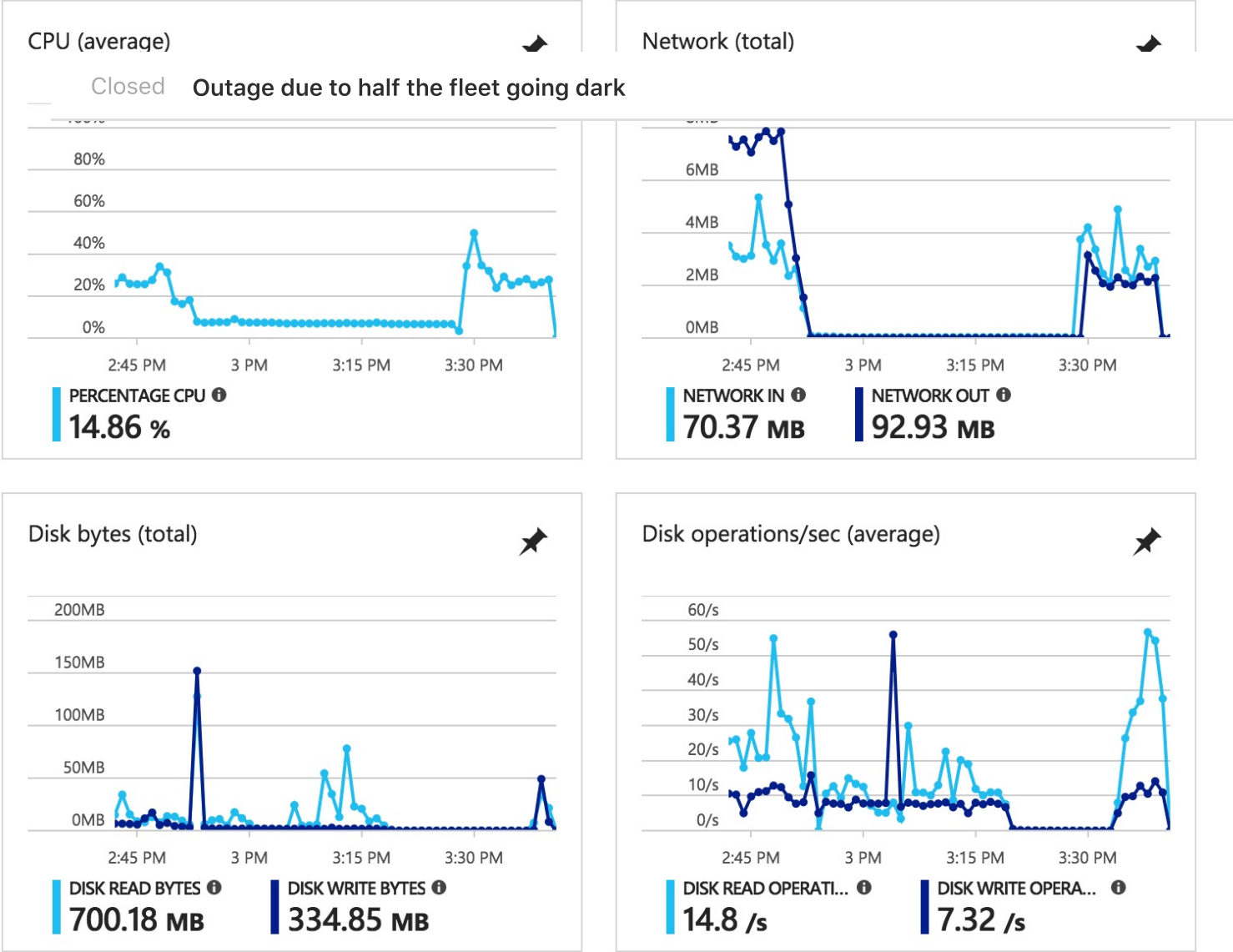




GC bad behavior:



Provider perspective:



## Root cause analysis

- GitLab.com started returning errors 502.
  - Why?
- Web-03, web-05, web-06, web-09 & web-10 disappeared from our monitoring and stopped processing web requests, web-01 was pegged with a load of 246
  - Why?
- We are still investigating why those web nodes went dark with our provider (cc [@victorcete](#)).
- Web-01 high load was due to `gitlab-pages` and `unicorn` processes locked on `D` state
  - Why?
- We are still investigating this with our provider, everything points at a networking communication problem with the NFS server for `gitlab-pages`


## Further actions

- Increase git GC stability by limiting resources on omnibus -> <https://gitlab.com/gitlab-com/infrastructure/issues/3141>
- Extract `gitlab-pages` out of the normal web fleet to isolate it on a specific set of nodes, remove the pages NFS mount from the web nodes
- Contain omnibus (runit) CPU and memory usage with cgroups to prevent taking a node completely in case the node got into thrashing mode.

Edited 2 years ago by [Pablo Carranza \[GitLab\]](#)

Linked issues ⓘ


0





Ilya Frolov


@ilyaf · 2 years ago

Mini version of this outage is what was yesterday: <https://gitlab.com/gitlab-com/infrastructure/issues/3135>.

- 

[Pablo Carranza \[GitLab\]](#) [@pcarranza-gitlab](#) changed the description 2 years ago
- 

[Pablo Carranza \[GitLab\]](#) [@pcarranza-gitlab](#) added [oncall](#) ~393244 labels 2 years ago
- 

[Pablo Carranza \[GitLab\]](#) [@pcarranza-gitlab](#) changed the description 2 years ago
- 



**Pablo Carranza [GitLab]** [@pcarranza-gitlab](#) changed milestone to [%WoW ending 2017-11-14](#) 2 years ago

Closed **Outage due to half the fleet going dark**



**John Jarvis** [@jarv](#) · 2 years ago

Owner

gitlab-pages processes are in D state everything points at a networking communication problem with the NFS server

Is it clear that the pages nfs mount was the root cause or was it all nfs mounts, possibly a networking issue, that caused these processes as well as the unicorn processes to be stuck in D state? Is there anything pointing to pages specifically besides the processes?



**Pablo Carranza [GitLab]** [@pcarranza-gitlab](#) · 2 years ago

[@jarv](#) I don't think that gitlab-pages is one to blame as the root cause, it all pointed more to a networking problem as there were also some unicorns in D state.

I do believe that by removing gitlab-pages from the regular web fleet we reduce the attack surface and isolate failure scenarios, that's why I'm suggesting going that way.



**John Jarvis** [@jarv](#) · 2 years ago

Owner

[@pcarranza](#) maybe we should prioritize nfsd probes that we aren't getting from the node exporter for the next oncall? Stats from nfsd would be nice, maybe with [https://github.com/arnarg/nfs\\_exporter](https://github.com/arnarg/nfs_exporter)?



**Pablo Carranza [GitLab]** [@pcarranza-gitlab](#) · 2 years ago

[@bjk-gitlab](#) 🙄 ?



**Ben Kochie** [@bjk-gitlab](#) · 2 years ago

Owner

Is this a problem you're trying to see form an NFS server or client side? We have NFS client and mountstats already enabled on our node\_exporters.

For servers stats, I have some work in progress to add /proc/net/rpc/nfsd to the node\_exporter .



**John Jarvis** [@jarv](#) · 2 years ago

Owner

[@bjk-gitlab](#) talking about nfsd specifically, would love to get some of [these stats](#).



**Pablo Carranza [GitLab]** [@pcarranza-gitlab](#) · 2 years ago

[@jarv](#) running that inside a container so we don't deploy any node.js?



**John Jarvis** [@jarv](#) · 2 years ago

Owner

[@pcarranza](#) yeah, for sure. May be worth doing until we get these metrics from the node exporter.



**Ilya Frolov** [@ilyaf](#) · 2 years ago

Over the weekend this was happening again.

web-0 [149] were not responding at all, web-06 had kernel thread locked in D state and load of 250. I had to reboot them from azure console, and restart gitlab-pages afterwards to make sure they return the correct certificate.



**John Jarvis** [@jarv](#) mentioned in issue [#3213 \(closed\)](#) 2 years ago



**John Jarvis** [@jarv](#) changed milestone to [%WoW ending 2017-11-21](#) 2 years ago



**Pablo Carranza [GitLab]**, @pcarranza-gitlab · 2 years ago

Closed   Outage due to half the fleet going dark



**Pablo Carranza [GitLab]**, @pcarranza-gitlab added `degradation` label 2 years ago



**Pablo Carranza [GitLab]**, @pcarranza-gitlab · 2 years ago

Created further action to increase visibility in our fleet going dark here: <https://gitlab.com/gitlab-com/infrastructure/issues/3248>

Closing this issue.



**Pablo Carranza [GitLab]**, @pcarranza-gitlab closed 2 years ago



**Andrew Newdigate**, @andrewn moved to [production#271 \(closed\)](#), 1 year ago

Please [register](#) or [sign in](#) to reply