


Closed (moved)

Opened 3 years ago by  [Alex Hanselka](#)

2017-06-15: Pages Outage Post Mortem

What happened?

- GitLab Pages were not being served up correctly, despite successful deploys.
- A successful deploy is defined to be a Pages pipeline for any given project completing with no errors.
- Not being served correctly is defined as returning 404s for new content OR not updating existing content with recent changes.
- During our corrective action, there was around 3-5 minutes of Pages giving 404 responses as it loaded all the domains.
- Currently the `gitlab-pages` daemon opens a listen socket before loading all the domains. This leads to 404s for any domains it has yet to load (see next steps section for improving this process).

Why?

- A change in mount points was made in order to migrate pages to their own NFS server ([#2004 \(closed\)](#).)
- The gitlab-pages daemon was not restarted, so it was serving things from the old mountpoint. Due to out of date docs and lack of familiarity with the daemon, it was not known that a restart was required.
 - A restart was required for the daemon to reload the mounts. Prior to a restart, it held open the old mount point, even though something else had mounted on top of it.

Timeline

- 6:30PM 2017-06-14: An rsync is started to migrate remaining pages data. This is expected to take 8 hours.
- 3:00AM 2017-06-15: The rsync is nearing completion but not yet finished, we decide to perform the cut over at this time.
- 3:15AM 2017-06-15: We begin rolling out the mountpoint change across the fleet.
- 3:40AM 2017-06-15: The mount point rollout is complete.
- 3:46AM 2017-06-15: The previous rsync finished and the final rsync is started. At this point all of the mount points are updated and the old server was expected to stop serving content.
- 7:39AM 2017-06-15: Reports come in about pages not being updated.
- 8:20AM 2017-06-15: We confirm that the mount points really are correct across the fleet.
- 8:45AM 2017-06-15: At this time we believe that the rsync is the culprit and it hasn't finished yet.
- 9:52AM 2017-06-15: The rsync has finished and pages updates are updating the correct NFS server, however the pages being served are still broken.
- 10:30AM 2017-06-15: No one with deep knowledge of gitlab-pages daemon is available to troubleshoot the issue in depth.
- 10:47AM 2017-06-15: An unrelated problem with an NFS server came up, bringing the application down, further slowing down Pages troubleshooting.
- 11:45PM 2017-06-15: The previous outage is resolved.
- 01:45PM 2017-06-15: We continue to troubleshoot and try to find someone with deep knowledge of the gitlab-pages daemon.
- 04:00PM 2017-06-15: It is determined that the gitlab-pages daemon is still serving from the old NFS server because it has not been restarted since the mount points changed.
- 04:08PM 2017-06-15: We begin restarting the gitlab-pages daemon across the fleet.
- 04:15PM 2017-06-15: The gitlab-pages daemon restarted across the fleet and the issue is now resolved.

What went well?

- GitLab Pages services were restored with zero data loss, only outage time.
- The vast majority (90+ percentile) were served correctly without outages or issues


What could be improved?

- It was hard to find people knowledgeable about gitlab-pages.
- Another outage happened during this one, diverting attention.
- Our gitlab-pages runbook doesn't speak to the conditions that we were observing.
- The only actionable item in the gitlab-pages runbook was not acted upon given no familiarity for what the outcome would be.
- We waited for an SME (Subject Matter Expert) to become available rather than following our documentation.

Next Steps?


- Document GitLab Pages process flow and implementation schemes so that Production Engineers understand what is happening and have a more comfortable sense with 'operating' on GitLab Pages (<https://gitlab.com/gitlab-com/infrastructure/issues/2041>)
- Bring up socket only after domains are loaded (<https://gitlab.com/gitlab-org/gitlab-ce/issues/33762>)
- Add a reload function to `gitlab-pages` so that we can attempt to reload without restarting the entire process. This may not help for this exact scenario, but it would help for troubleshooting efforts and possibly other unforeseen problems we might encounter (<https://gitlab.com/gitlab-org/gitlab-ce/issues/33763>)
- Act upon our runbooks, even if you're unsure.¹
- Get Pages through the production readiness questionnaire <https://gitlab.com/gitlab-com/infrastructure/issues/2071>


cc/ @gl-infra


1. In theory our runbooks should be peer reviewed and maintained, so even if the engineer directly handling the issue isn't as familiar with the subject they're troubleshooting - the guidance found within the runbook should be considered sound advice to be acted upon. 

Edited 3 years ago by [Pablo Carranza](#) [GitLab]


Linked issues   0

- 

[Alex Hanselka](#) [@ahanselka](#) mentioned in issue [#2035 \(moved\)](#) 3 years ago
- 

[Alex Hanselka](#) [@ahanselka](#) mentioned in issue [#2004 \(closed\)](#) 3 years ago
- 


[Alex Hanselka](#) [@ahanselka](#) changed the description 3 years ago




[DavChana](#) [@davchana](#) · 3 years ago

Old data served, but no loss for my pages, but experioenced & reported the issue @ Twitter <https://twitter.com/davchana/status/875403966133284864>

Now Sorted :)

- 

[Alex Hanselka](#) [@ahanselka](#) mentioned in issue [#2041](#) 3 years ago
- 


[Alex Hanselka](#) [@ahanselka](#) changed the description 3 years ago



[Alex Hanselka](#) [@ahanselka](#) · 3 years ago

I've added <https://gitlab.com/gitlab-com/infrastructure/issues/2041> to create more docs on Pages.

Owner



[Pablo Carranza](#) [GitLab] [@pcarranza-gitlab](#) · 3 years ago

[@ahanselka](#) I would like to continue the [blameless postmortem](#) following the [5 whys](#) and dig a bit further down to find a deeper root cause. I'll try to give you a bump to see where we go:

By how this is written down:

- The gitlab-pages daemon was not restarted, so it was serving things from the old mount point

Why wasn't the daemon not restarted?

Possible answers:

- It was overlooked in the process
- We haven't worked before with this daemon, so we don't know if it is necessary or not (documentation? testing?)

A different set of questions

Or another approach:

GitLab Pages were not being served up correctly, despite successful deploys

I would look into:

- defining what 页面无法访问 means.
- add the fact that we took an action to fix the issue and it resulted in pages returning 404 for roughly 20 minutes, and dive into this one.
- explaining a bit more the context of what 页面无法访问 means.

If by successful deploys means the users were pushing data and it was not showing on the public website, we need to understand how this disconnection happened, and what can we do to avoid finding ourselves in this situation (monitoring, process, documentation, code change?).

The serving 404s for that period of time has at least 2 ways to be looked at:

- The service needs to be changed to return a valid code until it can serve the contents (service unavailable?)
- The service needs to be changed to work correctly in a fresh start.
- We need to have some form of process in which we can roll a restart of this service without this form of disruption.

Does this make sense?



Alex Hanselka @ahanselka changed the description 3 years ago



Alex Hanselka @ahanselka · 3 years ago

Owner

@pcarranza it does indeed make sense. I've updated the description with more information. The changes are:

- Define what a deploy is (line 2).
- Define what 'not being served correctly' means (line 3).
- Add the period of 404s to the what happened with an explanation of why that happened.
- Addition of why the pages daemon was not restarted in the Why section.
- Corrective action with issue link for a more user friendly way to restart the pages daemon.
- Corrective action with issue link for a reload function to aid in troubleshooting.

Edited by **Alex Hanselka** 3 years ago



Pablo Carranza [GitLab] @pcarranza-gitlab · 3 years ago

@ahanselka

it was not known that a restart was required.

Why was a restart required?



Alex Hanselka @ahanselka changed the description 3 years ago



Alex Hanselka @ahanselka · 3 years ago

Owner

@pcarranza added to the description. The daemon was holding on to the old mount point, despite something else being mounted on top of it.

@ilyaf noted that /proc/<pid>/mounts will show you what mounts are currently known by a process. In this case, this was not the same as /proc/mounts , which is the the source of truth for what is mounted. Is this something we should monitor?

Edited by **Alex Hanselka** 3 years ago



Pablo Carranza [GitLab] @pcarranza-gitlab · 3 years ago

I created <https://gitlab.com/gitlab-com/infrastructure/issues/2071> as another following step to get Pages under control.



Pablo Carranza [GitLab] @pcarranza-gitlab changed the description 3 years ago



Pablo Carranza [GitLab] @pcarranza-gitlab · 3 years ago

Closing as we have further steps to continue with.



Pablo Carranza [GitLab] @pcarranza-gitlab closed 3 years ago



Andrew Newdigate [@andrewn](#) moved to [production#188 \(closed\)](#), 1 year ago

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