



[google \(/blog/tags.html#google\)](#)

[cloud native \(/blog/tags.html#cloud-native\)](#)

[GKE \(/blog/tags.html#GKE\)](#)

[kubernetes \(/blog/tags.html#kubernetes\)](#)

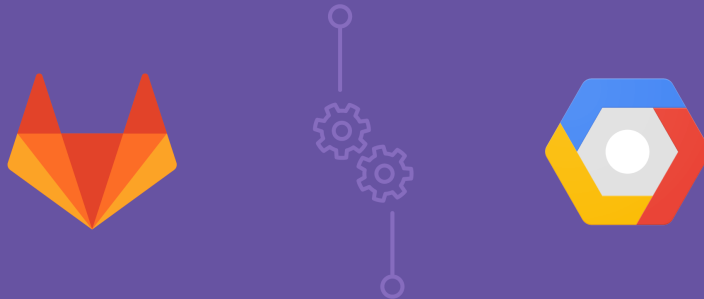
We're moving from Azure to Google Cloud Platform

Jun 25, 2018 · 5 min read · [Leave a comment](#)



[\(/company/team/#andrewn\)](#)

[Andrew Newdigate \(/company/team/#andrewn\)](#)



Update Jul 19, 2018: The latest info can be found in the [GCP migration update \(/blog/2018/07/19/gcp-move-update/\)](#) blog post.

Improving the performance and reliability of [GitLab.com \(/pricing/#gitlab-com\)](#) has been a top priority for us. On this front we've made some incremental gains while we've been planning for a large change with the potential to net significant results: moving from Azure to Google Cloud Platform (GCP).

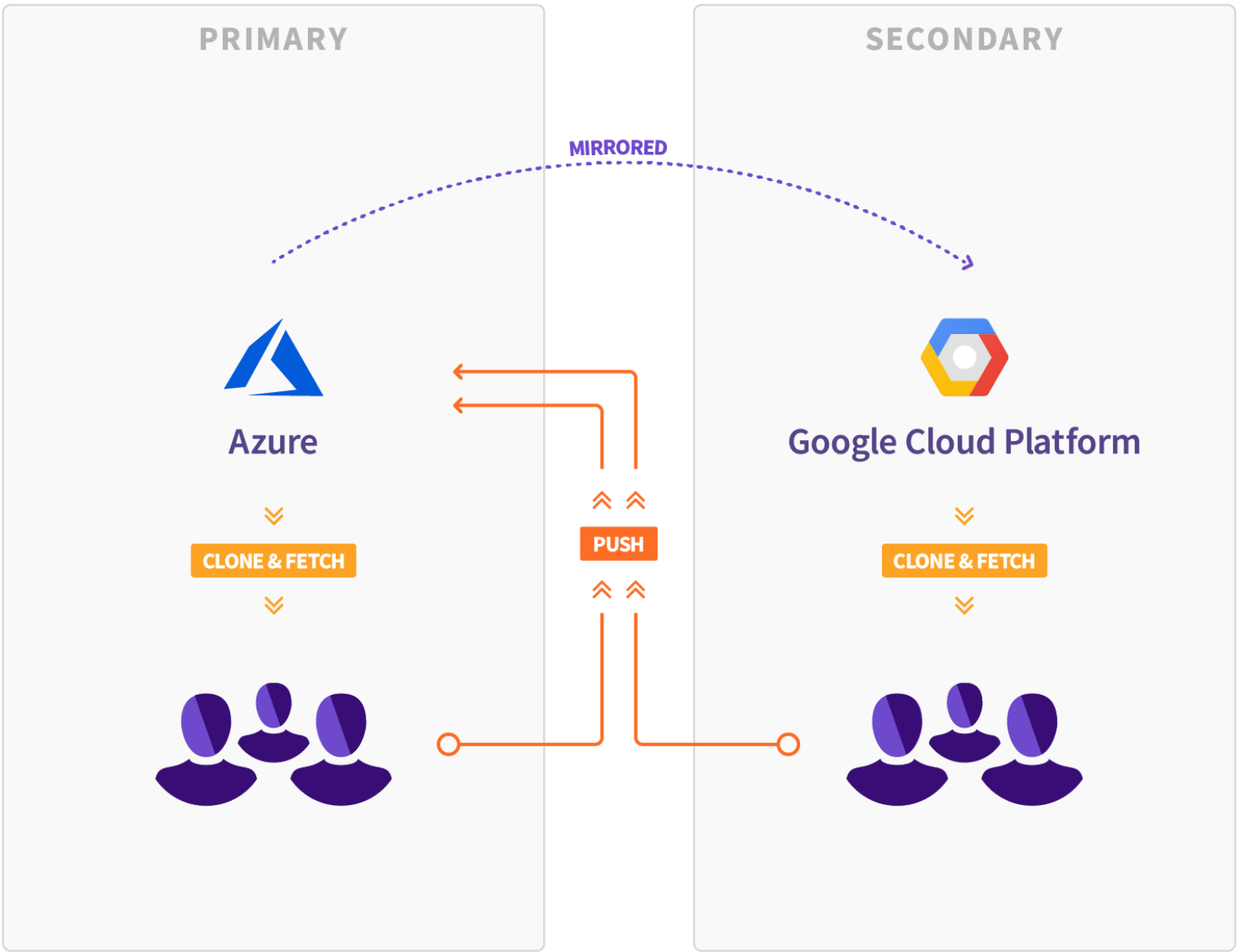
We believe [Kubernetes \(/solutions/kubernetes/\)](#) is the future. It's a technology that makes reliability at massive scale possible. This is why earlier this year we shipped native [integration with Google Kubernetes Engine \(/blog/2018/04/05/gke-gitlab-integration/\)](#) (GKE) to give GitLab users a simple way to use Kubernetes. Similarly, we've chosen GCP as our cloud provider because of our desire to run GitLab on Kubernetes. Google invented Kubernetes, and GKE has the most robust and mature Kubernetes support. Migrating to GCP is the next step in our plan to make GitLab.com ready for your mission-critical workloads.

Once the migration has taken place, we'll continue to focus on bumping up the stability and scalability of GitLab.com, by moving our worker fleet across to Kubernetes using GKE. This move will leverage our [Cloud Native charts \(https://gitlab.com/charts/gitlab\)](#), which with [GitLab 11.0 \(/blog/2018/06/22/gitlab-11-0-released/#cloud-native-gitlab-helm-chart-now-beta\)](#) are now in beta.

How we're preparing for the migration

One GitLab feature we are utilizing for the GCP migration is our [Geo product](https://about.gitlab.com/features/gitlab-geo/). Geo allows for full, read-only mirrors of GitLab instances. Besides browsing the GitLab UI, Geo instances can be used for cloning and fetching projects, allowing geographically distributed teams to collaborate more efficiently.

Not only does that allow for disaster recovery in case of an unplanned outage, Geo can also be used for a planned failover to migrate GitLab instances.



Following our mantra of dogfooding everything of our product, we are using Geo to move GitLab.com from Microsoft Azure to Google Cloud Platform. Geo is working well and scales because it's been used by many customers reliably since going GA. We believe Geo will perform well during the migration and plan this event as another proof point for its value.

Read more about Disaster Recovery with Geo in our [Documentation](https://docs.gitlab.com/ee/administration/geo/disaster_recovery/).

The Geo transfer

For the past few months, we have maintained a Geo secondary site of GitLab.com, called `gprd.gitlab.com`, running on Google Cloud Platform. This secondary keeps an up-to-date synchronized copy of about 200TB of Git data and 2TB of relational data in PostgreSQL. Originally we also replicated Git LFS, File Uploads and other files, but this has since been migrated to Google Cloud Storage object storage, in a parallel effort.

For logistical reasons, we selected GCP's `us-east1` site in the US state of South Carolina. Our current Azure datacenter is in US East 2, located in Virginia. This is a round-trip distance of 800km, or 3 light-milliseconds. In reality, this translates into a 30ms ping time between the two sites.

Because of the huge amount of data we need to synchronize between Azure and GCP, we were initially concerned about this additional latency and the risk it might have on our Geo transfer. However, after our initial testing, we realized that network latency and bandwidth were not bottlenecks in the transfer.

have on our Geo transfer. However, after our initial testing, we realized that network latency and bandwidth were not bottlenecks in the transfer.

Object storage

In parallel to the Geo transfer, we are also migrating all file artifacts, including CI Artifacts, Traces (CI log files), file attachments, LFS objects and other file uploads to [Google Cloud Storage \(https://cloud.google.com/storage/\)](https://cloud.google.com/storage/) (GCS), Google's managed object storage implementation. This has involved moving about 200TB of data off our Azure-based file servers into GCS.

Until recently, GitLab.com stored these files on NFS servers, with NFS volumes mounted onto each web and API worker in the fleet. NFS is a single-point-of-failure and can be difficult to scale. Switching to GCS allows us to leverage its built-in redundancy and multi-region capabilities. This in turn will help to improve our own availability and remove single-points-of-failure from our stack. The object storage effort is part of our longer-term strategy of lifting GitLab.com infrastructure off NFS. The [Gitaly project \(https://gitlab.com/gitlab-org/gitaly\)](https://gitlab.com/gitlab-org/gitaly), a Git RPC service for GitLab, is part of the same initiative. This effort to migrate GitLab.com off NFS is also a prerequisite for our plans to move GitLab.com over to Kubernetes.



Once or twice a week, several teams, including [Geo \(/handbook/engineering/development/enablement/geo/\)](/handbook/engineering/development/enablement/geo/), [Production \(https://about.gitlab.com/handbook/engineering/infrastructure/production/\)](https://about.gitlab.com/handbook/engineering/infrastructure/production/), and [Quality \(https://about.gitlab.com/handbook/engineering/quality/\)](https://about.gitlab.com/handbook/engineering/quality/), get together to jump onto a video call and conduct a rehearsal of the failover in our staging environment.

Like the production event, the rehearsal takes place from Azure across to GCP. We timebox this event, and carefully monitor how long each phase takes, looking to cut time off wherever possible. The failover currently takes two hours, including quality assurance of the failover environment.

This involves four steps:

- A [preflight checklist \(https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/preflight_checks.md\)](https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/preflight_checks.md),
- The main [failover procedure \(https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/failover.md\)](https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/failover.md),
- The [test plan \(https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/test_plan.md\)](https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/test_plan.md) to verify that everything is working, and
- The [failback procedure \(https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/failback.md\)](https://gitlab.com/gitlab-com/migration/blob/master/.gitlab/issue_templates/failback.md), used to undo the changes so that the staging environment is ready for the next failover rehearsal.

Since these documents are stored as issue templates on GitLab, we can use them to create issues on each successive failover attempt.

As we run through each rehearsal, new bugs, edge-cases and issues are discovered. We track these issues in the [GitLab Migration tracker \(https://gitlab.com/gitlab-com/migration/issues\)](https://gitlab.com/gitlab-com/migration/issues). Any changes to the failover procedure are then made as [merge requests into the issue templates \(https://gitlab.com/gitlab-com/migration/merge_requests?scope=all&state=all\)](https://gitlab.com/gitlab-com/migration/merge_requests?scope=all&state=all).

This process allows us to iterate rapidly on the failover procedure, improving the failover documentation and helping the team build confidence in the procedure.

When will the migration take place?

Our absolute [top priority \(https://gitlab.com/gitlab-com/migration#failover-priorities\)](https://gitlab.com/gitlab-com/migration#failover-priorities) for the failover is to ensure that we protect the integrity of our users' data. We will only conduct the failover once we are completely satisfied that all serious issues have been ironed out, that there is no risk of data loss, and that our new environment on Google Cloud Platform is ready for production workloads.

The failover is currently scheduled for Saturday, July 28, 2018. We will follow this post up shortly with further information on the event and will provide plenty of advance notice.

“We're moving @gitlab from Azure to @GCPcloud” – Andrew Newdigate

Click to tweet

40gitlab+from+Azure+to+%40GCPcloud%E2%80%9D+%E2%80%93+suprememoocow&url=https://about.gitlab.com/blog/2018/06/25/moving-

More to explore

 ENGINEERING

How to automate localization for Flutter apps

[\(/blog/2021/12/10/how-to-automate-localization-for-flutter-apps/\)](/blog/2021/12/10/how-to-automate-localization-for-flutter-apps/)

 ENGINEERING

GitOps with GitLab: How to tackle secrets management



Viktor Nagy

[\(/blog/2021/12/02/gitops-with-gitlab-secrets-management/\)](/blog/2021/12/02/gitops-with-gitlab-secrets-management/)

 ENGINEERING

GitOps with GitLab: Connect with a Kubernetes cluster



Viktor Nagy


[\(/blog/2021/11/18/gitops-with-gitlab-connecting-](/blog/2021/11/18/gitops-with-gitlab-connecting-)

[All Blog Posts → \(/blog/\)](/blog/)

Try all GitLab features - free for 30 days

GitLab is more than just source code management or CI/CD (<https://about.gitlab.com/topics/ci-cd/>). It is a full software development lifecycle & DevOps tool in a single application.

[Try GitLab Free \(/free-trial/\)](/free-trial/)



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Michael Bisbjerg • 3 years ago

So. I trust you keep copies of these checklists / todo lists / worklists off of [Gitlab.com](https://about.gitlab.com) as well. Would be bad if your migration stopped working halfway, only for you to be unable to access your checklists :).

Joking aside. This is an amazing feat. Do write more about it :).

4 ^ | v • Reply • Share ›

Andrew Newdigate → Michael Bisbjerg • 3 years ago

Thanks Michael, we will keep writing about it until the actual event.

You've noticed a very good point in fact regarding relying on [GitLab.com](#). This was a problem that was pointed out to us by someone on the team. We get around it by mirroring the project onto a second private GitLab instance and creating the actual issue on that server. This means that we will have access to the docs, no matter what happens.

2 ^ | v • Reply • Share ›

irregularshed • 3 years ago

So this has absolutely nothing to do with Azure being Microsoft and Microsoft buying Github?

2 ^ | v • Reply • Share ›

William Chia → irregularshed • 3 years ago

@irregularshed that is correct :) We've been planning this migration for a very long time while the MS acquisition is a relatively recent development. The migration is scheduled for July 28 and has required months of planning and practice to ensure a smooth transition. We would not be ready to do that if we had only decided to migrate once we learned of the acquisition.

3 ^ | v • Reply • Share ›

GitLab [GitLab](#) → irregularshed • 3 years ago

That's right.

^ | v • Reply • Share ›

nomadme → irregularshed • 3 years ago

I'm worried about Microsoft ruining another perfect product, when they dip their hands on Github...

^ | v • Reply • Share ›

nomadme • 3 years ago

Wow! I'm really glad you guys are moving out of Azure. You guys always rock and lead the pack!

Wish you luck on your huge move! Let me know if you need a helping hand packing boxes ;)

^ | v • Reply • Share ›

GitLab [GitLab](#) → nomadme • 3 years ago

We share the same enthusiasm. Here are some details after few months <https://about.gitlab.com/20...>

^ | v • Reply • Share ›

Kevin Urbainczyk • 3 years ago

Will this improve the general performance? (Loading pages, checkouts etc.)

^ | v • Reply • Share ›

Andrew Newdigate → Kevin Urbainczyk • 3 years ago

Kevin, this is not a goal of the "immediate" migration, although it is obviously a longer term goal and one of the reason we're moving to GCP and Kubernetes. Having said that we are cautiously optimistic that it will be a little faster from the outset.

1 ^ | v • Reply • Share ›

Ben Bodenmiller • 3 years ago

Where can one easily track the progress of moving GitLab off NFS? We also have that desire but it still seems to be required by GitLab app/API servers even when working with Gitaly.

^ | v • Reply • Share ›

Andrew Newdigate → Ben Bodenmiller • 3 years ago

Ben, almost all operations on [GitLab.com](#) are now running "NFS-free", going through Gitaly. What remains is a long-tail of low-frequency operations, and we are make good progress on finishing the last of these off.

The effort to remove all NFS operations from [Gitlab.com](#) is tracked through the Gitaly v1.0 milestone <https://gitlab.com/gitlab-o...>

The effort to remove all Rugged and file-system based git operations from the Rails application is tracked through the Gitaly v1.1 milestone: <https://gitlab.com/gitlab-o...>

So, once we reach Gitaly v1.0, [GitLab.com](#) will be able to run without NFS. Once we reach Gitaly v1.1, any GitLab instance will be able to run without NFS.

Some great metrics illustrating the state of migration are available in our public Grafana instance: look for Gitaly-tagged dashboards: <https://monitor.gitlab.net/...>

3 ^ | v • Reply • Share ›

Ben Bodenmiller → Andrew Newdigate • 3 years ago

Also, the "https://monitor.gitlab.net/" link is broken. The correct link is <https://monitor.gitlab.net/>

Also [https://monitor.gitlab.net/...](https://monitor.gitlab.net/) is great... how can I get the queries behind the graphs so I can add this to my Grafana w/ Prometheus setup?

^ | ▾ • Reply • Share ›

GitLab GitLab ➔ Ben Bodenmiller • 3 years ago

Hey Ben, have you read this blog post <https://about.gitlab.com/20...> It might be interesting for you. ^D

^ | ▾ • Reply • Share ›

Ben Bodenmiller ➔ Andrew Newdigate • 3 years ago

Can you help me understand why [gitlab.com](https://about.gitlab.com/) will be able to go NFS free after Gitaly v1.0 but other instances won't be able to until 1.1?

^ | ▾ • Reply • Share ›




 [Subscribe](#)  [Add Disqus to your site](#)[Add Disqus](#)[Add](#)  [Do Not Sell My Data](#)

 [\(https://www.facebook.com/gitlab/\)](https://www.facebook.com/gitlab/)  [\(https://twitter.com/gitlab/\)](https://twitter.com/gitlab/)  [\(https://www.linkedin.com/company/gitlab-com/\)](https://www.linkedin.com/company/gitlab-com/)

Sign up for GitLab's twice-monthly newsletter

☐ I would like to receive the bi-weekly newsletter via email



   [\(https://www.linkedin.com/company/gitlab-](https://www.linkedin.com/company/gitlab-)
[\(https://www.facebook.com/gitlab/\)](https://www.facebook.com/gitlab/) [channel/UCnMGQ8QHMA](https://www.youtube.com/channel/UCnMGQ8QHMA)

Why GitLab?

- [Product \(/stages-devops-lifecycle/\)](#)
- [Solutions \(/solutions/\)](#)
- [Services \(/services/\)](#)
- [DevOps tools \(/devops-tools/\)](#)
- [Is it any good? \(/is-it-any-good/\)](#)
- [Releases \(/releases/\)](#)

Resources

- [All resources \(/resources/\)](#)
- [All-Remote \(/company/culture/all-remote/\)](#)
- [Blog \(/blog/\)](#)
- [Newsletter \(/company/contact/\)](#)
- [Events \(/events/\)](#)
- [Webcasts \(/webcast/\)](#)

[Pricing \(/pricing/\)](/pricing/)

[Get started \(/get-started/\)](/get-started/)

[Topics \(/topics/\)](/topics/)

[Training \(/learn/\)](/learn/)

[Docs \(https://docs.gitlab.com/\)](https://docs.gitlab.com/)

[Developer Portal \(https://developer.gitlab.com/\)](https://developer.gitlab.com/)

[Install \(/install/\)](/install/)

Community

[Customers \(/customers/\)](/customers/)

[Contribute \(/community/contribute/\)](/community/contribute/)

[Community Programs \(/community/\)](/community/)

[Direction \(/direction/\)](/direction/)

[Technology Partners \(/partners/technology-partners/\)](/partners/technology-partners/)

[Channel Partners \(/partners/\)](/partners/)

[Open Source Partners \(/solutions/open-source/partners/\)](/solutions/open-source/partners/)

[GitLab for Open Source \(/solutions/open-source/\)](/solutions/open-source/)

[GitLab for Education \(/solutions/education/\)](/solutions/education/)

[GitLab for Startups \(/solutions/startups/\)](/solutions/startups/)

[Shop \(https://shop.gitlab.com\)](https://shop.gitlab.com)

[Community Forum \(https://forum.gitlab.com/\)](https://forum.gitlab.com/)

Support

[Get help \(/get-help/\)](/get-help/)

[Contact Sales \(/sales/\)](/sales/)

[Contact Support \(/support/#contact-support\)](/support/#contact-support)

[Support options \(/support/\)](/support/)

[Status \(https://status.gitlab.com/\)](https://status.gitlab.com/)

[Customers Portal \(https://customers.gitlab.com/\)](https://customers.gitlab.com/)

Company

[About \(/company/\)](/company/)

[What is GitLab? \(/what-is-gitlab/\)](/what-is-gitlab/)

[Jobs \(/jobs/\)](/jobs/)

[Culture \(/company/culture/\)](/company/culture/)

[Leadership \(/company/team/e-group/\)](/company/team/e-group/)

[Team \(/company/team/\)](/company/team/)

[Press \(/press/\)](/press/)

[Investor Relations \(https://ir.gitlab.com\)](https://ir.gitlab.com)

[Analysts \(/analysts/\)](/analysts/)

[Handbook \(/handbook/\)](/handbook/)

[Security \(/security/\)](/security/)

[Contact \(/company/contact/\)](/company/contact/)

[Terms \(/terms/\)](/terms/)

[Privacy \(/privacy/\)](/privacy/)

[Trademark \(/handbook/marketing/corporate-marketing/brand-activation/brand-standards/#trademark\)](/handbook/marketing/corporate-marketing/brand-activation/brand-standards/#trademark)

Git is a trademark of Software Freedom Conservancy and our use of 'GitLab' is under license