Migration to Google Cloud and Adopting Cloud Native with GitLab

GitLab Google Cloud

Why do some cloud migration initiatives quickly capture value while others fail?

Some fail because they attempt too much.

Others because they attempt too little.

Google Cloud and GitLab have each devoted significant resources into understanding the key factors that determine success or failure in cloud migration and application modernization.

In this eBook, we'll share vital insights that can help your organization achieve or exceed your goals for your journey to Google Cloud and offer actionable strategies and tactics that can help you succeed.

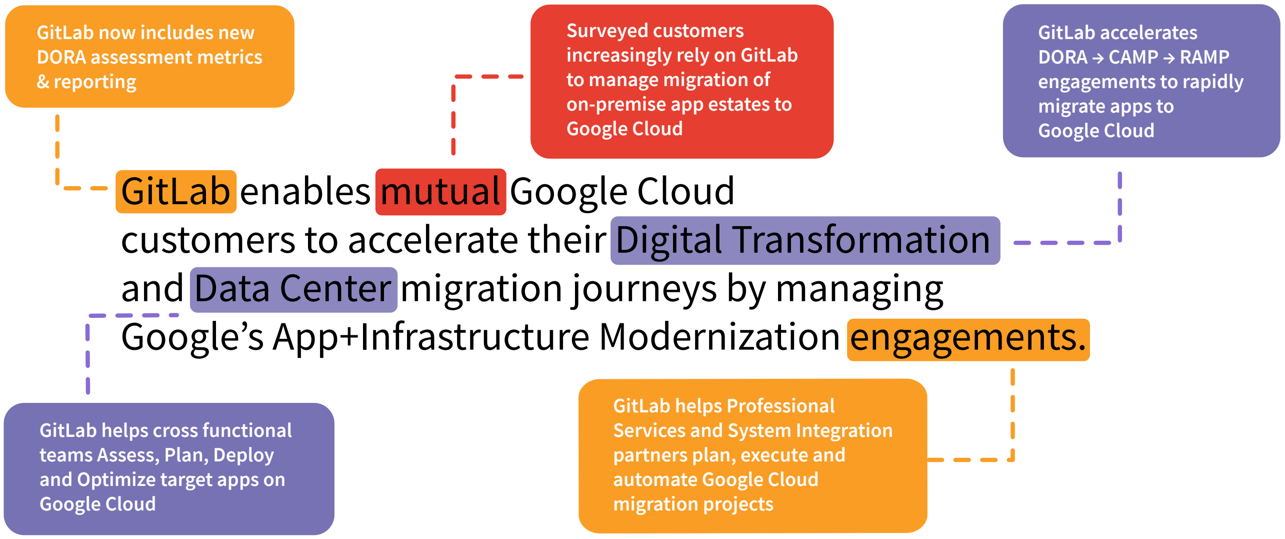
What is GitLab?

GitLab is The DevOps Platform that empowers organizations to maximize the overall return on software development by delivering software faster and more efficiently, while strengthening security and compliance. With GitLab, every team in your organization can collaboratively plan, build, secure, and deploy software to drive business outcomes faster with complete transparency, consistency and traceability.

GitLab's all-inclusive DevOps platform can help cross functional teams successfully Assess, Plan, Deploy and Optimize target applications onto Google Cloud.

Organizations using integrated Development and IT Operations tools or a consolidated DevOps platform like GitLab, will benefit by dramatically accelerating their Google Cloud adoption and can realize a significant financial return on their investment in five distinct areas:

- Software tool license cost reduction
- Eliminate tool chain integration costs
- Reduction in development costs
- Revenue acceleration from faster innovation
- Security and compliance risk mitigation



GitLab now includes new DORA assessment metrics & reporting

Surveyed customers increasingly rely on GitLab to manage migration of on-premise app estates to Google Cloud

GitLab accelerates DORA → CAMP → RAMP engagements to rapidly migrate apps to Google Cloud

GitLab enables mutual Google Cloud customers to accelerate their Digital Transformation and Data Center migration journeys by managing Google's App+Infrastructure Modernization engagements.

GitLab helps cross functional teams Assess, Plan, Deploy and Optimize target apps on Google Cloud

GitLab helps Professional Services and System Integration partners plan, execute and automate Google Cloud migration projects

Measurable operational results

GitLab's customers realized an 87% improvement in development and delivery efficiency time, a 12x increase in the number of annual releases for revenue-generating applications, and a reduction of code defects by a whopping 80%.[1]

Transform Your DevOps

Beginning a migration to Google Cloud presents the opportunity to assess your current DevOps approach and evolve it to embrace new approaches that offer more flexibility, agility, and reduces time to deploy. At GitLab, we identified four phases of evolution during the adoption of DevOps tools by organizations undergoing digital transformation over time.

Phase 1–Siloed DevOps
In this early phase, each department or team built or purchased their own tools in isolation.
Think: 5 different development teams using 5 different SCM solutions, or 3 different project teams using 4 different project management tools.

Phase 2–Fragmented DevOps
In this phase, organizations standardized on the same sets of tools across the Enterprise.
Think: All developers across the org using one SCM solution while all security engineers standardizing on one security solution.

Phase 3–DIY DevOps
Organizations tried to remedy fragmentation by manually integrating their DevOps point solutions.
Think: Optimization and maintenance overhead of integrating tools across a handful of standardized tools.

[1] *Source: The Total Economic Impact™ Of GitLab, a commissioned study conducted by Forrester Consulting, June 2020*

Phase 4–The DevOps Platform Era
The DevOps Platform is a single application with one user interface and a unified data store. Think: GitLab as a single tool for all software development with an integrated user experience for interdisciplinary teams across Dev, Sec, and Ops.

Organizations that embrace the DevOps Platform approach see major improvements in deployment frequency, lead time for changes, mean time to restore, change fail rate, and other key metrics. To quickly capture your investment in Google Cloud, organizations need to embrace available GitOps and DevOps workflows and processes, optimized for cloud as well as other existing architectures, as the central anchor of the cloud migration and modernization journeys.

The Journey to Google Cloud with GitLab

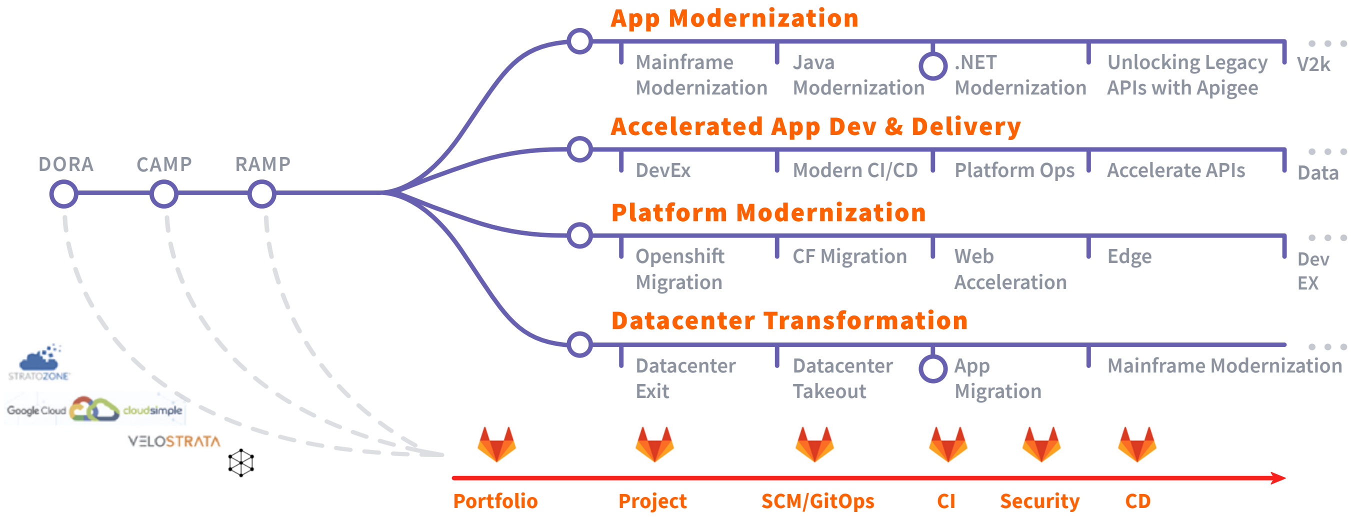
The first step in planning a cloud migration is determining the type of migration aligned to your cloud journey roadmap:

1. **Lift and Shift:** Landing non-refactored, non-cloud-native workloads, translated 1-1 from existing architecture to the Cloud, that enable immediate OpEx cost basis benefits for the business. (Ex. VM or database migration.)
2. **Improve and Move:** Modernizing the workload while migrating it through partial refactoring and/or expanding functionality of existing workloads with adoption of native, cloud managed services. (Ex. Replacing applications with managed cloud services.)
3. **Rip and Replace:** Decommission an existing application and completely redesign and rewrite it as a cloud-native app. (Ex. Building new services with AI/ML).

Each migration strategy has different benefits, considerations, and investments, but all routes to cloud are only as successful as the business' ability to adopt the right Google Cloud services for tomorrow's SaaS like platforms. Regardless of strategy, GitLab can help lower the barrier of each step of adoption throughout the migration journey as defined below:

1. **Assess:** Dedicated assessment and discovery of your existing infrastructure environments, application inventory, and cost of ownership. Include a deep dive into the current state of security and operational practices from development to deployment.
2. **Plan:** Establish cloud infrastructure requirements for your workloads and their corresponding execution strategies for migration.
3. **Deploy:** Implement and execute a deployment process to move workloads into the cloud infrastructure using DevOps best practices.
4. **Optimize:** Post deployment, iterate to continually expand your business's potential by incrementally improving performance, scalability, disaster recovery, costs, training, etc.

GITLAB IS THE DEVOPS PLATFORM FOR MIGRATION TO THE GOOGLE CLOUD



GitLab pairs well with Google's existing migration solutions, including:

ASSESS Current State

Application migration projects start with an assessment to rationalize which apps to migrate, modernize and/or replace. It is very likely, as part of your migration, some workloads will be lifted and shifted, some modernized, and some replaced.

The DORA report for example can serve as an assessment framework by identifying key metrics to calculate productivity and estimate the potential ROI of your transformation initiatives. Leveraging these metrics can help you increase your capabilities and improve your IT performance as you progress along the migration and modernization journey.

To enhance DORA metrics, businesses consolidating their DevOps tooling and enabling collaboration into a common platform like GitLab can get better insights and visibility on areas of excellence or improvements.

For example GitLab customers using the breadth of the product from planning, managing, developing to deploying can begin to collect key DORA metrics across their SDLC concerning Deployment Frequency, Lead Time, Mean Time To Recover, Change Failure Rate, Value Streams and Flow.

Continually tracking and measuring these indicators solves important pain points and is a crucial step in providing a closed feedback loop for continual improvement through constant assessment.

PLAN for Successful Migration

In this phase, organizations take the outcomes of the assessment after they've identified the apps and desired approach to migrate, modernize and/or replace. Now you can begin executing tasks required to provision and configure the cloud infrastructure and services that will support migrating workloads onto Google Cloud. Building a foundation of critical configurations and services is an evolving process that requires flexibility.

Plan to establish your rules, governance, and settings at an abstract level; avoid making decisions that lock you into a specific way of doing things. Think implementing frameworks, not prescriptive recipes.

Non-Disruptive Innovation

A Cloud Center of Excellence (CCoE) can be one approach to piloting concurrent DevOps practices. A CCoE starts by creating a CCoE leadership team of diverse, knowledgeable experts from across the organization to define best practices for the rest of the cloud team to follow. With a pioneering team, best practices can slowly be established, codified, templatized, and incrementally adopted by other groups within the organization.

> **Having a CCoE or interdisciplinary team utilize a GitOps approach on the GitLab platform gives the organization the ability to practice "Non-Disruptive Innovation."**
>
> **The value of non-disruptive innovation is that the team can test new workflows, processes, and services outside of daily development, operations, and security projects without impacting existing business services.**

Mobilizing people successfully means new ways to approach organization and work. Take for example GitOps, which requires advance planning and close collaboration between cross-functional teams to automate and bridge the gap between the lifecycle of infrastructure and services.

Infrastructure as Code (IaC)

Standardization through IaC empowers teams through self-service, allowing infrastructure to be codified, version controlled and automatically provisioned. Cloud operations adopting IaC gives the ability to build and test without the delays of provisioning and manual intervention queues.

As with application code, infrastructure configurations are updated through merge requests, enabling transparent and access-controlled changes to be stored in GitLab instead of bespoke config files and desired state stored offline in hidden folders on custom hardware, firewall or network access configurations.

Building Blocks for Cloud Native

In addition to operational planning, cloud adoption requires careful consideration of the core building blocks being employed as part of the new architecture in the cloud.

For example, plan for scalability and cloud networking services for *lift and shift* workloads that can benefit from a range of compute sizes, regional and geographic disbursement for availability purposes, and even intelligent auto-scaling capabilities in the cloud.

Anthos Migrate could be one of those solutions enabling batch migration of VM-based services into the cloud. Anthos Migrate in conjunction with GitLab in such instances can also be codified and automated to help templatize or abstract a desired workflow for bulk migration.

Similarly, if migration is part of a multi-cloud / hybrid cloud initiative, consider designing architectures that leverage solutions like Google Cloud Anthos that inherently provide capabilities to scale various types of compute across heterogeneous infrastructures. Designs that are container first and cloud native can yield a level of abstraction and flexibility regardless of where they deploy workloads.

GitLab supports deploying workloads on Anthos across CloudRun, GKE, and provides compatibility from an operational perspective as well with Anthos Configuration Management to promote GitOps like deployment methodologies.

DEPLOY workloads to Google Cloud

After building a foundation for your target Google Cloud environment, you can begin to deploy your workloads. Microservice-based cloud native solutions should not require large major deployment releases. Instead there should be a focus on small but frequent iterative updates. The best performing organizations are able to push multiple code updates to services daily versus the past annual or bi-annual updates.

With a ‘unified DevOps’ approach, GitLab has enabled organizations through consolidating diverse capabilities into a single application across all the stages of DevOps. GitLab features and capabilities can address the needs of all stakeholders in the software lifecycle from security, operations, Q/A, development, to project management groups leveraging a single user experience.

Moving in this direction starts by changing the deliverable mindset.

GitOps takes DevOps best practices for development and applies them to infrastructure automation. In short, GitOps delivers a hermetic, immutable process (Ex.GitLab and Anthos Config Management as demonstrated in this video).

At GitLab, we see GitOps principles applied to all kinds of different infrastructure and automation, from old-school VMs to containers and Kubernetes clusters as well.

OPTIMIZE

Many organizations are benchmarking against the DevOps Research and Assessment (DORA) findings. DORA has developed four key metrics that the industry has widely adopted as performance indicators for software development teams:

1. **Deployment frequency:** How often an organization successfully releases to production.
2. **Lead time for changes:** The amount of time it takes for code to reach production.
3. **Change failure rate:** The percentage of deployments that cause a failure in production.
4. **Time to restore service:** How long it takes an organization to recover from a failure in production.

Since GitLab is collecting the relevant information from your teams as they work, on demand reporting of these metrics can help each team accelerate its efforts by measuring how organization changes can influence productivity that leads to continual deliveries.

The goal of the optimize stage of the journey is to provide the foundation for the next iteration. Continual improvement requires continual action to address performance metrics throughout the software development to deployment lifecycle.

Standardizing on metrics, being able to collect them, and finally taking action steps is the core principles of optimization.

In summary, GitLab and Google Cloud present tightly aligned planning methodologies, migration strategy frameworks and tools to help you successfully Assess, Plan, Deploy and Optimize target applications into the Google Cloud.

> **For more information or to request a demo, email us at: google-alliances@gitlab.com or visit https://about.gitlab.com/partners/technology-partners/google-cloud-platform/**
>
> **Interested in purchasing through Google Cloud Marketplace with integrated cloud billing? Read here for more information.**