Migrate and Modernize with GitLab and AWS

A Guide to Application Modernization on AWS

GitLab | aws

Your organization has made the decision to migrate to Amazon Web Services (AWS).

Migrating and modernizing applications en masse to the cloud is challenging at best–a concern shared by 80%[1] of CIOs who admit their own cloud adoption initiatives haven't delivered on the expected business agility.

Development and IT Operations teams are being challenged to deliver and manage cloud migrations to enable innovative applications to be released faster. Innovation enables their organizations to attract and keep customers while reducing the complexity and costs of their existing IT ecosystem.

Innovation is achieved when these teams apply Agile practices and principles to organize and manage their work, whatever their chosen methodology. In this eBook, we outline how to leverage the AWS migration methodology in conjunction with a GitOps framework in which DevOps teams collaborate to create an agile development ecosystem. We will demonstrate how GitLab's all-inclusive DevOps platform helps these teams successfully Assess, Mobilize, Migrate and Modernize target applications to the AWS cloud. We will discuss how GitLab enables all stakeholder teams to communicate, collaborate, and automate the software development lifecycle. The GitLab platform provides the framework and tooling to manage, plan, create, verify, package, secure, release, configure, monitor, and protect the entire application stack. GitLab also allows cloud and app migration teams to further leverage their current toolchain with integrations into external issue trackers and other Continuous Integration/ Continuous Development (CI/CD) tools.

Organizations using integrated Development and IT Operations tools or a consolidated DevOps platform like GitLab, will benefit by dramatically accelerating their AWS Cloud migration 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

Measurable operational results

The results speak for themselves. 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%.[2]

In collaboration with Amazon Web Services (AWS), we have developed this guide of best practices and insights for migrating business applications and modernizing the software delivery strategy with a product-focused delivery model.

[1] *Source: McKinsey Study: Unlocking business acceleration in a hybrid cloud world, July 2019*

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

Opportunity to Reinvent

A new cloud migration initiative presents the opportunity to reinvent team dynamics, company culture and internal processes, as well as improve security and compliance requirements by adopting a transparent and auditable GitOps approach.

> **It's difficult to bridge disparate methodologies and siloed toolchains to plan, prioritize, automate, and track new migration projects when confronting a jumble of mobile, distributed, cloud native, legacy, and mainframe apps.**

Compounding these challenges are legacy application development workflows, processes, and tools that are poorly suited to the needs of cloud architectures and hybrid infrastructures.

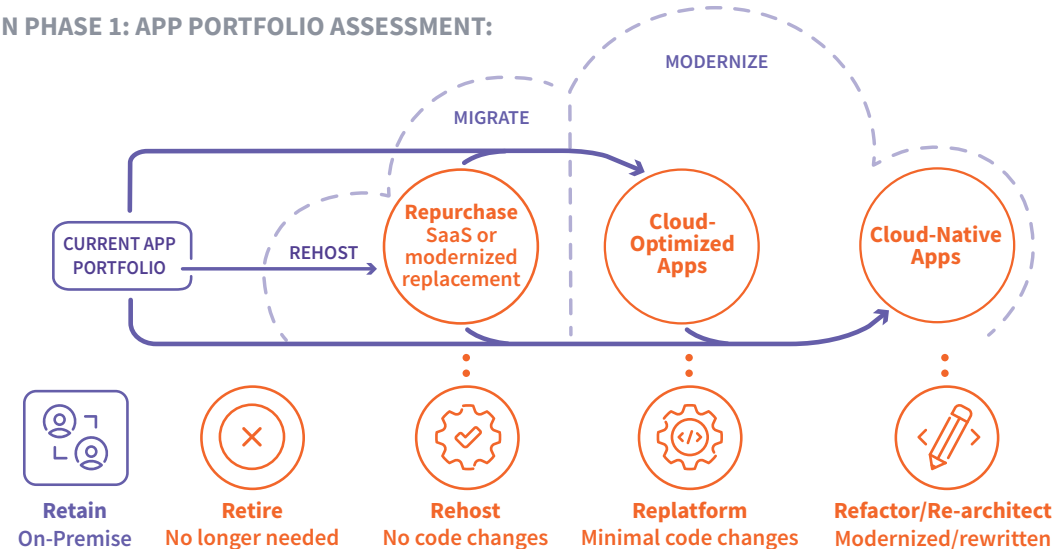
To capture the full value of the AWS cloud, organizations need to embrace modernized GitOps and DevOps workflows and processes, optimized for cloud as well as other existing architectures, as the central anchor of the cloud migration and modernization journeys.

Assess, Mobilize, Migrate and Modernize

The first step in planning a cloud migration is to follow the AWS recommended three-phase migration process:

1. **Assess**
2. **Mobilize**
3. **Migrate and Modernize**

MIGRATION PHASE 1: APP PORTFOLIO ASSESSMENT:



ASSESS Cloud Adoption Capabilities

Application migration projects start with an assessment to rationalize which apps to migrate and/or modernize.

There are many models including the Gartner TIME and Forrester's Wave Methodology.

The AWS best practices for migrations starts with a 6Rs approach to understand the complexity of migration patterns. In this methodology, there are six different migration patterns:

1. **Retain**–There are some applications that are not suitable for migration to AWS. This can be due to data sovereignty or an existing application architecture that may preclude migration to a virtual environment. These applications can be left in place.
2. **Retire**–Applications are no longer useful to the business and can be retired.
3. **Rehost**–Applications are migrated to AWS without making any change to the components (application code, database, or other components) of the application.
4. **Replatform**–The applications being migrated to AWS involve software version upgrades or configuration changes to adapt to the new architecture platform on AWS.
5. **Repurchase**–If the current version of the application is not optimized for deployment on AWS, or the current terms and conditions don't allow a bring-your-own-license (BYOL) model to AWS, moving to a Software as a Service (SaaS) model or replacing with modern off-the-shelf software is advisable.
6. **Refactor/Re-architect**–The application is re-architected and rebuilt with a new code base to utilize cloud native features. This migration pattern is used primarily when the business objective is to improve agility, scalability, and performance of the application that is otherwise not achievable with the current architecture.

Template Rationalization Assessments

To improve speed and reliability, rationalization assessments can be templated into a list of discrete and repeatable steps for each app with owners (assignees) and stakeholders (participants) identified as "issues" in GitLab Manage.

This enables the migration team to inventory their application estate, then manage migration and modernization initiatives as a collection of projects in GitLab through epics, groups (programs), and milestone dates for organizing and tracking progress. GitLab's included monitoring and value stream metrics can augment the quality of management reporting.

MOBILIZE for Successful Migration

A Cloud Center of Excellence (CCoE) is a customary approach to piloting concurrent DevOps practices. Cross-disciplinary "2-pizza teams" allow early adopters and tiger teams to implement the governance, best practices, training, and architecture needed for cloud adoption in a manner that provides repeatable patterns for the larger enterprise to follow."[3]

A CCoE provides the structure for the teams tasked to remove complexity and risk to enable faster delivery of high-quality software at a lower cost.

The DevOps and GitOps approach to Mobilizing

AWS defines DevOps as "the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes." At its core, DevOps is a cultural change: removing the barriers between traditionally siloed teams of developers, security, operations, and database. With a DevOps approach, we break down those barriers and have them working together, from the beginning in parallel, not linear tracks. With GitLab's DevOps platform, each of these groups becomes part of a cohesive, collaborative team owning the responsibility of the service or application, not just a component.

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

How the GitLab DevOps Platform enables companies to adopt a modern approach to DevOps and GitOps principles:

Package

GitLab enables teams to package their applications and dependencies, manage containers, and build artifacts with ease.

Configure

GitLab helps teams to configure and manage their application environments. Strong integration to Kubernetes reduces the effort needed to define and configure the infrastructure required to support your application.

Secure

GitLab provides Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Container Scanning, and Dependency Scanning to help you deliver secure applications along with license compliance.

Monitor

Get feedback and the tools to help you reduce the severity and frequency of incidents so that you can release software frequently with confidence.

Release

GitLab helps automate the release and delivery of applications, shortening the delivery lifecycle, streamlining manual processes, and accelerating team velocity.

Protect

GitLab provides cloud native protections, including unified policy management, container scanning, and container network and host security.

[3] *Source: "Cloud Enablement Engine: A Practical Guide" AWS, July 2021*

MIGRATE and MODERNIZE

To capture the full value of their cloud investments, organizations need to embrace modernized DevOps workflows and processes, optimized for the cloud.

By incorporating the GitLab DevOps platform with a cultural shift to modern agile development, teams are empowered to work together at the same time.

The GitLab approach to Concurrent Development enables organizations to deliver software much faster and more efficiently, while strengthening security and compliance.

GitLab does this by providing a single unified platform for all stages of the migration and software development lifecycle. The advantage of a single tool is that all stakeholders from the practitioners to the executive team are working in a single tool, using a single taxonomy.

Getting started with GitLab doesn't mean immediately getting rid of your existing Dev and Ops toolchain. GitLab integrates easily with a wide range of popular applications.

As your teams identify the workloads and applications that are candidates for Rehost, Replatform and ReFactor, GitLab becomes the central hub for those migration programs.

From a leadership and management perspective, it will allow you to automatically understand the progress your teams are making with GitLab Value Stream Metrics.

GitLab Value Stream Analytics allows for one shared view of the team's velocity. With insights into how long it takes the team to move from planning to monitoring, it points to bottlenecks in the development process, enabling management to uncover, triage, and identify the root cause of slowdowns in your migration programs giving your team the data to successfully meet your goals.

Benchmarking

Many organizations are benchmarking against the DevOps Research and Assessment (DORA) findings. DORA has developed four key metrics that the industry has widely adopted that you can use 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 your team accelerate its efforts by measuring how organization changes can influence productivity that leads to fast deliveries.

Migration Templates

With the Assess and Mobilize phases completed, your teams can start using GitLab to build migration templates (collection of tasks and deliverables assigned to accountable parties) in GitLab.

GitLab Plan enables portfolio planning and management through epics, groups (programs), and milestones to organize and track progress. When moving from planning to execution, GitLab will provide your teams with Value Stream Metrics.

Breaking down your organization's internal silos can increase velocity, but it is challenging in larger organizations with multiple competing processes, tools and cultures.

A key step is to start with a single source of truth for all software and GitOps configurations with **GitLab Create**–GitLab helps teams design, develop, and securely manage code and project data from a single, distributed version control system to enable rapid iteration and delivery of business value.

The second element of velocity is automation, which is key to removing information and process barriers between traditionally siloed teams of developers, security, operations, database. Validate and automate migration actions with **GitLab Verify**.

GitLab helps AWS cloud development, delivery, operations, and security teams fully embrace continuous integration to automate the builds, testing, integration, and verification of their code.

Automation is also key to **GitLab Package**, **GitLab Secure**, **GitLab Release**, and **GitLab Configure**.

As Ops, compliance, and security teams develop self-service components (pipelines, test, IaC), they can safely test and deploy new features based on governance requirements.

Infrastructure as Code

Standardization through IaC allows teams to manage infrastructure like code, with version control and automated testing. Templates and testing tools support governance and compliance models, making environments more secure and auditable. It will also allow teams to develop self-service infrastructure and the ability to build and test without the delays of provisioning queues.

By then leveraging **GitLab Protect** as part of those pipelines, you can automate the enforcement of your organization's compliance and governance standards. GitLab Protect prevents security intrusions using unified policy management, container scanning, container network and host security with configurable alerting rules. App dependencies such as database and YML configuration files can be stored as code in GitLab.

When preparing to deploy the rehosted applications to the AWS cloud, users can securely manage and update their Terraform or Cloud Formation Templates/CDK in **GitLab Release** and **GitLab Configure** as GitOps scripts. As with application code, GitOps configurations are updated through merge requests, enabling transparent and access-controlled changes to be stored in GitLab instead of bespoke config files stored offline in hidden folders on custom hardware, firewall or network access configurations. Improving DevOps performance requires transparent monitoring of KPIs. **GitLab Monitor** tracks CI/CD pipeline events and job information to gain deep, granular insights on throughput and test performance over time. Operational insight is key to shipping code faster and more efficiently.

Summary

Migrations are complex. Your organization may be complex and siloed. But your organzation's future is dependent on these migrations to be more competitive, nimble and resource optimized.

DevOps offers a proven methodology for accelerating an organization's ability to deliver faster and become more competitive in the marketplace. Managing Infrastructure as Code through GitOps can be a game changer for organizations migrating to AWS.

Rethink the Toolchain

> Take advantage of CCoE. Leverage automated migration and management tools. Work together from a single application, sharing a single codebase, datastore, installation, interface, overview, and workflow–**proven to be 200% faster than the traditional DevOps lifecycle.**

The GitLab DevOps platform and workflow ensure accountability and that code is security tested prior to deploying into production.

Iterate faster and innovate together

The goal is to remove complexity and risk providing everything necessary to deliver high quality, more secure software faster at a lower cost.

GitLab is a purpose-built DevOps Platform that provides the visibility, accountability, governance, and velocity to enable all teams to work more efficiently together.

GitLab will:

1. Increase Operational Efficiencies.
2. Deliver Better Products Faster.
3. Reduce Security and Compliance Risk.

> **Ready to Start Your Journey?**
>
> **For more information or to request a demo, email us at: aws-sales@GitLab.com or visit https://about.gitlab.com/partners/technology-partners/aws/**