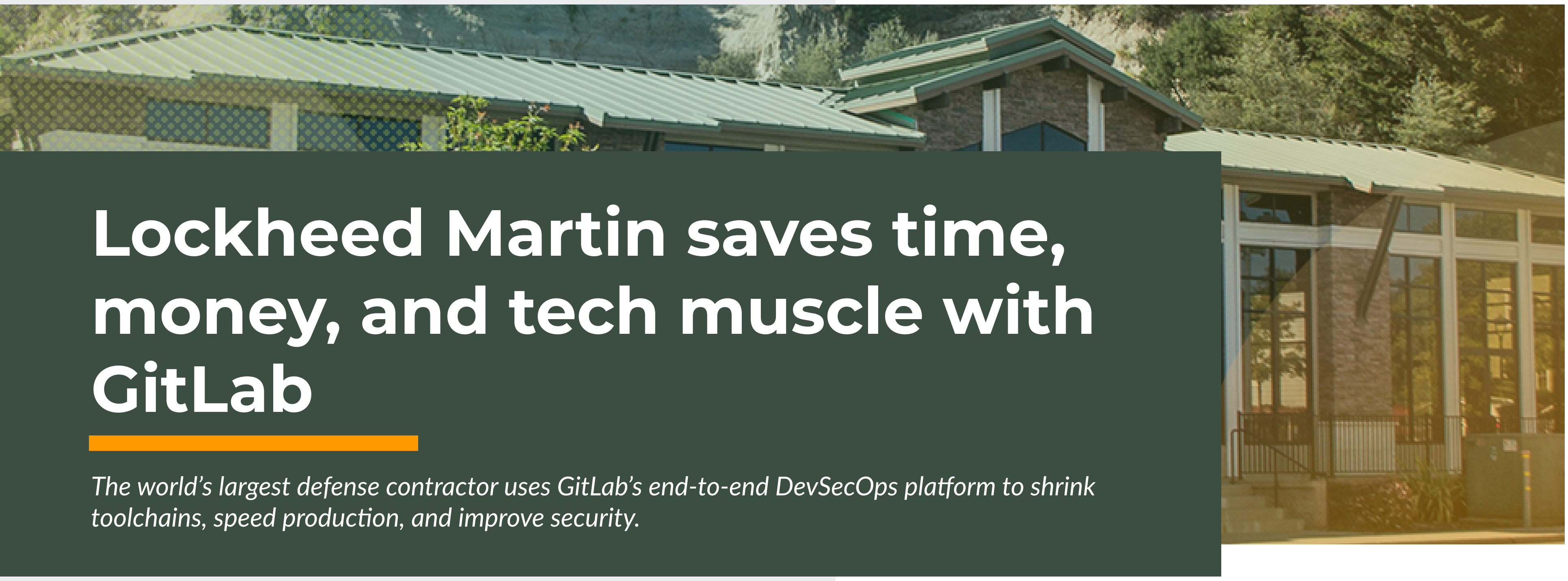


Lockheed Martin saves time, money, and tech muscle with GitLab

The world's largest defense contractor uses GitLab's end-to-end DevSecOps platform to shrink toolchains, speed production, and improve security.

Lockheed Martin Corp., an American aerospace, defense, information security, and technology giant, has adopted GitLab's single, end-to-end DevSecOps platform to more efficiently, securely, and quickly develop and deploy software for thousands of their programs, ranging from satellite platforms and aerospace systems to ground control software and maritime surface and subsurface software.

> BY SWITCHING TO GITLAB AND AUTOMATING DEPLOYMENT, TEAMS HAVE MOVED FROM MONTHLY OR WEEKLY DELIVERIES TO DAILY OR MULTIPLE DAILY DELIVERIES.
>
> ***- Alan Hohn,***
> *Director of Software Strategy, Lockheed Martin*

ACHIEVING MISSION NEEDS WITH SPEED AND FLEXIBILITY

Based in Bethesda, Maryland, Lockheed Martin has approximately 116,000 employees worldwide, with more than 370 facilities. The company is principally engaged in the research, design, development, manufacture, integration, and sustainment of advanced technology systems, products, and services. The majority of the company's business is with the U.S. Department of Defense and U.S. federal government agencies. In the aeronautics industry alone, Lockheed Martin had $26.7 billion in sales in 2021.

Lockheed Martin's customers depend on the company to help them overcome their most complex challenges and to stay ahead of emerging threats by providing the most technologically advanced solutions. Their engineering teams need speed and flexibility to meet the specific mission needs of each customer, while using shared expertise and infrastructure to ensure affordability.

TOOLCHAIN COMPLEXITY

Lockheed Martin has a history of using a wide variety of DevOps tools — everything from ClearCase to Jenkins, Dimensions, Redmine, and Bitbucket. Each program or product line at the company had its own toolchain. Rarely was one like another, with team members simply picking tools they were familiar with.

That led to uneven efficiencies and results.

Alan Hohn, director of software strategy at Lockheed Martin, said the quality of a team's development and deployment environment often was based on how lucky the DevOps team happened to be. Well-funded programs, or those with forward-leaning leadership, might have had high-quality automation for everything from testing to continuous deployments, while other programs might only have had a build server to run compiles. And in the worst cases, programs might not have had any automation at all, creating more hands-on work, using team members' time, and increasing the chance of missed problems.

80X FASTER CI PIPELINE BUILDS

THOUSANDS OF JENKINS SERVERS RETIRED

90% LESS TIME SPENT ON SYSTEM MAINTENANCE

INCREASING COLLABORATION

To add to the challenge, software development teams at Lockheed Martin had made numerous attempts to set up code repositories that would allow developers to reuse code across programs — but these repos were rarely, if ever, used because they were never incorporated into the multitude of environments where teams were actually developing the software. That meant developers, without a solid collaboration environment, were always starting from scratch and the code in the repositories simply "sat and rotted," says Hohn.

Once Lockheed Martin widely adopted GitLab's platform, software sharing and reuse easily became part of their day-to-day operations.

"Having GitLab has completely changed the way we approach reusable software because the place where we develop software is also the place that other people can share, contribute, and participate in that development," says Hohn. "Now, all of our programs have access to a high-quality software development environment."

This development environment translates directly to benefits for Lockheed Martin's customers. One team working on a program for the U.S. Department of Defense was able to reduce build times from 12 hours to 4 hours through the use of GitLab pipelines with containerized builds. This enabled the program to run 16 builds per night, rather than four, increasing test frequency and software quality. Overall, build success went from 60% to 90%. As a result, the customer is receiving new capabilities sooner and with better quality.

CREATING CONTINUITY

One of the most significant challenges Lockheed Martin faced in driving collaboration between programs and across the entire business is that their software resided in many different systems, with differing security requirements. GitLab, together with Lockheed Martin's Software Factory, enables the company to modularize their software so reusable components can be shared in globally accessible environments, while teams can still maintain tight control over software components that are mission-critical or have security constraints.

A key feature of Lockheed Martin's Software Factory is that it provides a common GitLab CI configuration YAML and common CI container images that are pre-configured to work with their other software development tools.

Teams have to maintain separate environments for security reasons, so their configuration needs to be able to work in different hosting environments for image registries and associated tools. They also often need to sustain particular versions of their software for years, since sometimes it's deployed into operational environments where it can't be frequently updated.

To take on these challenges, Hohn's team created a catalog of common pipelines for popular programming languages with modules for security scanning, container image builds, and semantic versioning. The pipeline catalog allows developers to use the same YAML source file in multiple environments without having to make modifications. The catalog also makes it possible to create a historic build using a specific version of a pipeline and guarantees version consistency.

Before adopting GitLab, Lockheed Martin had made a few attempts to build a common set of pipelines, but they ended up only supporting a few users and were overly prescriptive. It wasn't enough.

With GitLab as the foundation for their pipeline catalog, Lockheed Martin has been able to create pipeline templates that can be reused in multiple environments, including disconnected networks. That saves time and effort, and ensures continuity among projects. Now, pipeline updates are simpler, and sophisticated testing and release management are easier and more efficient.

"Now, we can be confident changes to our pipelines are automatically and thoroughly tested, and we can easily support both fast-moving development teams and risk-averse teams that are maintaining mission-critical capabilities," says Hohn. "Our new approach, built on some key GitLab CI features, has helped us find the right balance of commonality and customization."

Lockheed Martin now serves 2,500 pipelines per minute through the common pipeline catalog.

DELIVERING SCALE

Because rapid adoption of GitLab created the need for a more scalable solution, Lockheed Martin, GitLab, and Amazon Web Services (AWS) worked together to automate and optimize Lockheed Martin's code deployment across the enterprise. The solution started with a well-architected review of the design between the three companies. AWS then helped to automate and optimize the Lockheed Martin GitLab deployment for the CI/CD environment by delivering Infrastructure as Code (IaC) to deploy the environment in two hours, instead of the several hours it had taken previously.

The AWS team also established workflows to deliver a fully automated, highly available Disaster Recovery architecture for GitLab that is compliant and scalable. This enabled a consistent process that runs without manual intervention. AWS also supported load balancing to auto-scale the deployment process based on developer demand for pipeline runs and user traffic so developers are not waiting on deployments to execute. Pre-migration testing was performed to establish baselines, followed by post-migration testing to measure performance and scalability gains in delivering faster deployments.

Additionally, monitoring and security controls were implemented to comply with Lockheed Martin's security policies. As a result, the team was able to deliver operational efficiencies with the number of build requests waiting to be processed dropping from 200 to zero, and reducing time for code deployment across the enterprise. This effort showcased how large enterprises with thousands of software developers can build and deploy automated, scalable, and resilient code pipelines in the cloud using platforms such as GitLab by leveraging AWS best practices.

GOING WITH GITLAB

Lockheed Martin didn't need to do a formal evaluation of GitLab's platform before deciding to go with it. The company's DevOps teams have, over the years, used a multitude of tools on the market, so they understood the capabilities and benefits they would get from GitLab. For instance, GitLab's built-in continuous integration capability was a "killer feature," according to Hohn.

And the company has gone in big with GitLab. Today, they have approximately 64,000 projects on the GitLab Platform — some legacy projects have been migrated to GitLab, and others were initiated on the platform.

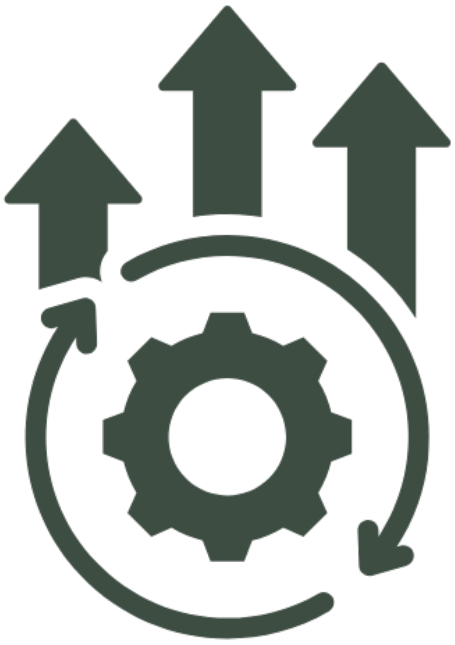
Lockheed Martin hasn't rid itself of all of its toolchains but the company has greatly reduced them, cutting complexity, cost, and workload.

"We recognize that there are going to be programs where the customer wants to own the development environment and they want a specific tool," explains Hohn. "We want to get to the point, and what we're achieving, is people don't even consider standing up their own toolchain. They just use GitLab because they know that it works.

"Have we erased all traces [of toolchains]? No," adds Hohn. "But it's so small it's not significant to us."

For instance, before widely adopting GitLab, the company was using Jenkins for CI servers – thousands of instances of Jenkins across the organization. That's no longer the case.

"Everybody had to maintain a different installation. We've collapsed that down to mostly GitLab," says Hohn. "There are still Jenkins instances out there but they're a small fraction of the number there were three years ago."



SAVING TIME AND EFFORT

Those reduced toolchains are saving the company time, muscle, and money, so that it can continue to innovate and deliver affordable solutions to their customers.

“For teams that had independent environments, they had to dedicate about 20 hours a week and 80 hours a month just to keep the system running,” says Hohn. “On a team of 12 people, that’s dedicating at least half a person. We reduced that by something like 90%. They’re now spending a couple of hours of a person’s time. That’s multiplied across a lot of teams. As a corporation with more than 10,000 software engineers, we can say we’re saving hundreds or thousands of hours a year.”

Having all of those projects on GitLab’s single platform means legacy programs that might have previously, on average, delivered to testing monthly and to operations quarterly, now are delivering to testing every six days and to operations every 26 days, Hohn notes.

“It’s very typical that we’re seeing monthly deliveries turn into weekly,” he adds. “We’re seeing quarterly deliveries turn into monthly. I mean, that scale of changes is very common.”

That time savings means the company can respond to customer requests more confidently and reliably.

“We serve a lot of customers and we have a lot of software development activities,” says Ian Dunbar-Hall, senior software engineer at Lockheed Martin. “GitLab allows a team to go from zero to a repository and a full CI pipeline, totally self-service, in 30 minutes, rather than the 40 hours, minimum, that it used to take.”

IMPROVING SECURITY

Since Lockheed Martin works with the Department of Defense and federal agencies, the company builds systems that are critical to national security. That means creating secure software is integral to both Lockheed Martin and its customers. A challenge for any company using toolchains is that it’s easy to miss an update because of the sheer size and complexity of the chain. Now with GitLab, they don’t have to worry about using tools that haven’t been updated because with a single, end-to-end platform, an update only has to be done once and every instance is covered.

And with defense and security-focused customers, compliance is a critical issue for Lockheed Martin. That’s easier to manage now that the company uses GitLab’s compliance framework to enforce software quality, and automation to make releases and dependency management more efficient and faster.

Using a single platform also means teams are getting a standardized set of automated security capabilities — from cutting-edge analysis tools to vulnerability scanning and security automation — seamlessly built in. Before using GitLab, teams didn’t all have the best security tools and there was no standardized way of handling security practices. Now with Lockheed Martin’s common pipeline catalog, teams also are using off-the-shelf pipelines that already have best-in-class security built in. “Now that we have a more common approach, it’s much easier for teams to leverage a common way of doing software build, test, and security scans, which raises the level of quality of the products that we create,” says Hohn.

“Today, it is extremely common to have sophisticated security scanning capabilities as part of all of our pipelines because the effort involved to add that to the pipelines is so much smaller,” says Hohn. “Teams now are aware of the security posture of the code they’re writing in a way that they weren’t before. That enables conversations about the security of our software that were not taking place the old way.”

The company is still using some third-party, legacy security tools, but teams are using GitLab’s platform to make it easier to integrate them all. “It’s a great complement,” says Jeff Daniels, Director Automations and Applications. “It’s easier now that we have GitLab, which increases our security posture and quality.”

Lockheed Martin is looking to continue to grow with GitLab. DevOps teams will migrate even more of their projects over to the DevSecOps platform and expand from there. “We hope to see growth in the number of projects using security and software supply chain functionality, including compliance pipelines and dashboards,” says Hohn.