Case Study

How Siimpl Reduced Build Times by 90% with Docker Build Cloud and GitHub Actions

Prepared by Siimpl, a leader in cloud-first solutions and engineering innovation.

About Siimpl: Siimpl delivers efficient, high-quality software solutions for forward-thinking businesses. We specialize in microservice architecture, API design, and automation and have expertise in modern cloud/on-prem environments. Our world-class developers excel in the continuous delivery and integration of infrastructure with proficiency in a multitude of technologies and platforms. With a core team of former Microsoft and Amazon engineers, we offer the agility of a startup and the experience of a large firm, ensuring seamless integration and innovative solutions.

Learn more at siimpl.io or contact solutions@siimpl.io

Highlights

90%
faster build time

Achieved by replacing Docker Buildx and QEMU emulation with Docker Build Cloud and self-hosted GitHub runners, reducing multi-architecture build times by 90% compared to the previous configuration.

50%
reduction in lead time

Improved DORA metrics for Lead Time for Changes and Time to Restore by approximately 50%.

30%
faster MTTD & MTTR

Shortened Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR) by around 30%, with potential improvements up to 50-60% through further optimization.

> "Docker Build Cloud helped us cut build times by 90%. Before, we were stuck dealing with QEMU emulation, which slowed everything down, but now, it's a night-and-day difference."
>
> **Neal Patel**
> Siimpl

Table of Contents

Introduction

A leading East Coast-based cybersecurity company faced severe performance bottlenecks as it scaled, which threatened its ability to meet growing client demands. Siimpl, a cloud-first solutions provider, assisted the cybersecurity company in implementing solutions to improve its DORA (DevOps Research and Assessment) metrics, such as Lead Time for Changes and Time to Restore.

The cybersecurity firm, known for protecting digital assets, faced several operational inefficiencies. The root of these challenges lay in outdated infrastructure and misaligned development environments. As the company grew, its systems struggled to handle increasing demand, leading to disjointed environments, slow build processes, and unreliable incident response mechanisms that were not designed for the complexity of their expanding operations.

The journey from identifying problems to implementing solutions highlights strategic innovations and technical expertise. Neal Patel from Siimpl explains, “We managed to turn these obstacles into stepping stones, enhancing our overall performance and stability.”

> “Docker Build Cloud’s remote builders have been incredible. Developers don’t have to worry about their local machine specs anymore—they just offload the builds to Docker, and builds are completed much faster."
>
> **Neal Patel**
> Siimpl

Challenges

Overcoming operational hurdles in cybersecurity

The cybersecurity company faced several challenges that hindered its engineering operations and threatened growth. Leadership realized the urgency for addressing these operational blockers to maintain their competitive edge and ensure robust performance for their clients.

Efficient development and deployment

One major issue was more synchronization between developers' environments and the CI/CD pipeline. This disconnect made it difficult for engineers to confidently test changes, resulting in a slow and cumbersome deployment process. The deployment process was slow, with each deployment taking up to several hours to complete. Engineers often faced delays due to testing inconsistencies across development environments, causing bottlenecks in getting code from development to production. As a result, development, QA, and operations teams were all affected, leading to frustration and reduced productivity.

The misalignment negatively affected key DORA metrics, notably the Lead Time for Changes and Time to Restore, which are critical for measuring development efficiency. The company needed a seamless and consistent environment across development, testing, and production stages to streamline its workflow and boost productivity.

Reliable incident response

Another significant challenge was the inability to roll back to stable versions during deployment issues quickly. The company's existing container infrastructure was not optimized for efficient rollbacks. Without a properly automated container versioning system, reverting to stable builds required manual intervention, slowing down incident response and extending downtime.

In the chaotic moments following a deployment problem, the team struggled to revert to previous stable versions swiftly, which threatened their goal of achieving 99.99% uptime. Establishing a robust infrastructure for quick and efficient rollbacks was essential to maintaining system reliability and minimizing downtime.

Comprehensive telemetry

The third challenge revolved around telemetry and observability. Despite efforts from the Site Reliability Engineering (SRE) team to implement telemetry collection and publishing, the tools in place were ultimately not effective due to low adoption. The system relied on fragmented and outdated tools that required too much manual setup, discouraging developers from fully integrating them into their workflows.

Consequently, the company faced delays in detecting and resolving issues, increasing risk for their business and clients. To address this, they needed to standardize telemetry configuration and simplify the setup of auto-instrumentation libraries. This would improve developer experience and enable actionable alerts to reduce Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR).

Solution

The right tools for optimized operations

To address the cybersecurity company's challenges, Siimpl implemented strategic solutions centered around Docker Build Cloud and GitHub Actions. These targeted interventions streamlined development workflows, reduced build times, stabilized incident response reliability, and improved telemetry and observability across the organization.

CI/CD configuration with self-hosted GitHub runners and Docker Build Cloud

Initially, the company's CI/CD pipeline relied on Docker Buildx and QEMU to emulate different architectures, significantly slowing down build times. The implementation was further improved with the adoption of Docker Build Cloud. Optimizing local builds is bottlenecked by the developers' local chip architecture, but Build Cloud allows the Docker engine to seamlessly integrate with remote builders. This allows developers to take advantage of native architecture build speeds with minimal overhead.

"With Docker Build Cloud, we don't have to worry about local hardware holding us back. Developers can build and test natively across different architectures, and it just works," Patel says. Find specific details on implementation in Siimpl's GitHub repository.

Leveraging SemVer-tagged containers for easy rollback

The unpredictable nature of deployments often required quick rollbacks. Siimpl introduced a Semantic Versioning (SemVer) tagging strategy to manage container images. This approach enabled the company to quickly revert to previous stable versions when issues arose, minimizing downtime. DevOps teams configured automated jobs using AWS CLI commands to update Amazon Elastic Container Service (Amazon ECS) services with the desired image tags, ensuring quick recovery and minimal operational disruption.

Configuring sidecar containers in Amazon ECS for aggregating and publishing telemetry data

Addressing the telemetry challenges, Siimpl used Terraform modules to embed extensive configuration into the client's infrastructure. Sidecar containers were defined in Amazon ECS task definitions to run OpenTelemetry (OTel) collectors, which aggregated and published telemetry data from application containers. This setup decoupled the telemetry collector from the runtime container, ensuring application stability even during telemetry failures.

Additionally, multi-stage builds configured in Dockerfiles were used to standardize the initialization of auto-instrumentation libraries across the client's Node.js microservices, resulting in clean and efficient images.
"We configured sidecar containers to ensure our main applications remained stable, even if telemetry systems encountered issues," Patel says.

Task definition example

By implementing multi-stage builds, the team was able to tackle varied build processes efficiently. These Dockerfiles separated the build environment from runtime, ensuring images were clean and optimized. The process involved installing OpenTelemetry libraries during the build and copying them during runtime, providing a consistent, reliable workflow across applications.

Key benefits

The solutions implemented by Siimpl addressed the cybersecurity company's challenges by introducing several key features. These features resolved immediate issues and laid the groundwork for a more efficient and robust engineering operation.

90% faster build times

from switching to Docker's native node strategy with self-hosted GitHub runners, eliminating the performance lag caused by Docker Buildx and QEMU emulation.

Instant rollbacks and minimized downtime

from implementing Semantic Versioning (SemVer) for container images, enabling quick reverts to stable builds with automated recovery through AWS Command Line Interface (AWS CLI).

Increased system stability

during telemetry failures from decoupling telemetry from runtime containers using Amazon ECS sidecar containers and OpenTelemetry collectors, improving system observability and health monitoring.

Cleaner and more efficient images

from using multi-stage Dockerfiles, which separate build and runtime stages, standardizing auto-instrumentation across the company's Node.js microservices.

"With Docker Build Cloud, our developers can stay in their local terminals, and builds happen seamlessly without needing to push code or create a pull request. It's all done right in Docker Desktop, which is a massive time saver."

Neal Patel
Siimpl

Results

Achieving operational excellence

The solutions implemented by Siimpl resolved the key technical challenges and drove noticeable improvements across the business and development teams. These changes resulted in faster development cycles, more reliable systems, and smoother operations.

One of the most substantial impacts was the 90% reduction in local build times, achieved by switching to Docker Build Cloud remote builders and self-hosted GitHub runners. Engineers could now confidently test and deploy code across multi-architecture environments, freeing them from the previous delays caused by Docker Buildx and QEMU emulation. The accelerated builds meant development teams faced fewer bottlenecks and faster iterations in the CI/CD pipeline, translating to faster delivery of new features to customers.

Implementing Semantic Versioning (SemVer) for container images made it possible to quickly revert to stable versions during deployment issues, which had been a significant source of downtime. For the business, the ability to minimize downtime — helping the company achieve its goal of 99.99% uptime — improved service reliability and reduced the risk of negative customer impact. "With Docker's containers, automating rollbacks became so much easier," Patel explains. "Now, when there's an issue, the team can recover quickly without a lot of manual work, which really cuts down on downtime." Additionally, the AWS CLI automation further streamlined the recovery process, making the rollback strategy efficient and reliable.

Using sidecar containers and OpenTelemetry collectors, Siimpl helped improve telemetry and system observability, which was previously fragmented. Developers now had real-time insights into system health, enabling faster detection and resolution of issues. The result was a 30% reduction in Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR), with the potential for further optimization up to 50-60% through alert tuning and automation. Improved observability meant fewer incidents and faster recovery times, resulting in less customer disruption and more proactive system monitoring. "Using Docker for versioning has saved us so much time. We can roll back to stable versions almost instantly, which is important to stay at 99.99% uptime," Patel says.

The impact of these changes is invaluable across the organization. Customers are happier with the pace at which feature requests are released. The product team has gained more confidence in the engineering team thanks to the improved rollback strategy and targeted alerting. Engineers are excited about the ease of instrumentation for observability and the improved build times.

Contact Sales

> "One of the best things about Docker Build Cloud is that it reduces the overhead. Our team doesn't have to wait for a GitHub Actions job to finish. Everything happens natively in Docker Desktop, which keeps the workflow fast and efficient."
>
> **Neal Patel**
> Siimpl

> "The developer productivity gains we've seen by using Docker Build Cloud are massive. We're talking about a 50-75% improvement in cycle time for builds, which is a game changer for our team."
>
> **Neal Patel**
> Siimpl

> "The performance gains we've seen using Docker Build Cloud have been huge, particularly with multi-architecture builds. Developers now have access to fast, native builds without the delays caused by emulation."
>
> **Neal Patel**
> Siimpl