docker buildcloud

# Empower Your Development Teams

## Overcoming Inner Loop Bottlenecks for Peak Productivity

# Introduction

In today's fast-paced digital landscape, software development teams face significant challenges that can impede their productivity and innovation. The inner loop—the iterative process of writing, testing, and debugging code—is a crucial environment where your team aims to work efficiently and embrace creativity throughout their build process. However, this crucial phase is often plagued by challenges that hinder productivity. Common pain points such as slow build times, frequent context switching, and environment inconsistencies can lead to significant project timeline delays, frustrated developer teams, and increased costs.

## Understanding the Inner Loop

If you lead a software development team, you already know that empowering your developers to do their best work pays off with more efficient workflows and faster release cycles. Provisioning an environment where cloud services are leveraged alongside programmers' preferred local tools is a recipe for happy developer teams and boosted inner loop development performance.

Your team almost certainly spends most of its time working in the inner loop. Since we moved from mainframes to PCs for programming, developer teams have been using the inner loop to build code on a local machine. Developers prefer working in the inner loop because it helps them stay in the flow and focus on completing tasks without interruptions. Enabling your team with efficient inner loop processes is critical for their overall development workflow, impacting everything from the quality of the initial code writing to its resilience during deployment.

When developers push their code to version control, it moves to the outer loop. In the outer loop, everything else happens that leads up to release. This includes code merge, automated code review, test execution, deployment, canary release, and observation of results. Today, the outer loop typically includes an automated CI/CD pipeline, making the outer loop as fast, efficient, and automated as possible.

> **The inner loop with its local development environments remains crucial for developers.**

A reliable PC or Mac enables developers to quickly write, build, and test code on their machines before deploying it to production. However, traditional local-only development can be inefficient, as resources may fall short of the demands of modern applications, leading to slow build times and limited cross-team collaboration.

# Common Frustrations in the Inner Loop

Developer teams frequently encounter several pain points within the inner loop, including:

**Slow build times:** Long waits for builds to complete can significantly delay development. An Incredibuild study found that most developers reported that build time had increased an average of 15.9% in 2021, and another Incredibuild study found that they spend an average of 57 minutes a day waiting for builds to finish. This isn't a solid block of time wasted waiting, but the accumulation of slightly longer waits on repetitive builds throughout the day. Time, whether it's developers' or compute hours, is money. According to the 2023 Stack Overflow Developer Survey, the average programmer costs $83 an hour. Every hour spent waiting for builds is real money down the drain. Long build times are also annoying. A 2023 survey of C++ developers found that 81% considered slow build times a pain point, with 43% finding it a major pain point. Unhappy developers aren't productive programmers.

**Frequent context switching:** Shifting between tasks breaks concentration and slows progress. Continuous interruptions prevent developers from achieving deep focus and creative breakthroughs.

> A UCLA study found that every time a developer switches tasks, it takes an average of 23 minutes and 15 seconds to get back on track.

These frustrations not only slow down development but also lead to developer dissatisfaction and increased costs. For instance, a small company with 25 developers earning an average salary of $140,000 could face potential costs of around $420,000 annually due to slow build times and frequent context switching.

# How to Overcome Inner Loop Bottlenecks

Even a small reduction in build time leads to productivity improvements. A recent Google study published in IEEE Software, 'Build Latency, Predictability, and Developer Productivity,' showed that "every change to build latency can increase the likelihood of developers staying on task." This means that when a build is faster, developers are more likely to stay in flow and retain their task's context. If a build is too slow, they will task switch away and perform less well.

Docker Build Cloud addresses these common inner loop frustrations and enhance developer productivity and streamline workflows by:

**Speeding up build times:** One of the most significant benefits of Docker Build Cloud is its ability to offload builds to the cloud, drastically reducing the time developers spend waiting for builds to complete. This allows developers to focus more on coding and less on waiting, enhancing productivity and reducing costs. Docker Build Cloud can accelerate Docker image builds up to 39 times faster than traditional methods.

**Simplifying tool management:** Build Cloud eliminates the need for finicky emulators by supporting native AMD and ARM builds. This removes the complexity of managing separate builders and streamlines the development process, allowing developers to work more efficiently.

**Reducing context switching:** By streamlining the build process and providing a consistent environment, Docker Build Cloud reduces the need for frequent context switching. This consistency allows developers to stay in the flow and retain their task's context. By reducing build times, Docker Build Cloud helps developers reclaim this lost time, further enhancing productivity.

**Enhancing collaboration:** A shared build cache enables team members to access cached results, reducing redundant builds and speeding up the development cycle. This ensures that all team members and CI/CD pipelines use the same cached layers, leading to more reliable builds. This feature is particularly beneficial for teams working on the same repository. Shared build caches help teams avoid repetitive work and promotes collaboration. By reusing previously built layers or outputs, developers can iterate faster and avoid the "it works on my machine" problem.

**Docker Build Cloud can significantly accelerateCI builds by offloading them to cloud-based BuildKit instances, reducing build times by up to 70%. This is particularly beneficial for companies with extensive CI/CD pipelines, as it allows developers to focus more on coding and less on waiting for builds, leading to significant improvements in build efficiency and reduced resource usage.**

We know how important it is for your teams to maintain efficiency and consistency without disrupting existing workflows. Docker Build Cloud integrates smoothly with existing development tools and workflows, such as Docker Compose and GitHub Actions. This seamless integration means teams can adopt these solutions without overhauling their current processes. Build Cloud supports hybrid local/cloud development, allowing developers to use familiar local environments while leveraging the cloud for resource-intensive tasks.

Whether your teams are building locally or in a continuous integration pipeline, you can adopt Docker Build Cloud without replacing your current processes. This approach enables leveraging familiar local development tools, such as editors and debuggers, while scaling to cloud resources for demanding tasks, deployments, or collaboration.

**By ensuring seamless integration with your current tools, Docker Build Cloud empowers your teams to achieve higher productivity and efficiency without any disruptive changes.**

By cutting down build times, your company will save money. For example, as mentioned above,

**small organizations can save around $420,000 annually by reducing build times with Docker Build Cloud. These savings can be reinvested into other business areas, promoting growth and innovation.**

# Real-World Success Stories with Docker Build Cloud

The following success stories showcase how Docker Build Cloud's powerful features have a proven track record of delivering tangible business value, driving productivity, reducing costs, and enhancing collaboration for leading organizations, and using real work repos.

**Switchboard: Streamlining Web3 Development**

Switchboard, a company focused on supercharging Web3 applications, faced significant challenges with cross-compiling Docker images for multiple platforms. Docker Build Cloud provided a native multi-architecture solution that eliminated the need for slow emulators, dramatically improving build times. The shared cache feature further enhanced their development process, enabling rapid iteration and faster delivery of new features. Switchboard CTO DoctorBlocks highlighted, "Our overall build times improved considerably through the shared cache feature. Before, on our local machine, it took 15-20 minutes to build; now, with Docker Build Cloud, it's down to 110 seconds, a massive improvement."

**Fast, simplified multi-platform builds**

**Posthog: Accelerating Innovation**

Posthog, an open-source product analytics platform, leveraged Docker Build Cloud to reduce build times by 9.6x. This improvement allowed their developers to focus more on innovation and less on waiting for builds to complete. The seamless integration with their existing CI/CD pipeline further streamlined their development workflow.

**Seamless CI/CD integration**

**Mastodon and Moby Projects**

Both prominent in the open-source community, the Mastodon social network and Moby container projects have seen substantial improvements in build times. Docker engineers achieved Docker image build speeds up to 39x faster with Mastodon, and up to 36% faster with Moby. These gains translate into more efficient development processes and quicker release cycles, enhancing the ability to respond to user feedback and market demands.

**up to 39X faster builds**

**Enterprise Collaboration Software**

One of Docker's technology customers, who develops enterprise collaboration software, was able to reduce their build time from an average duration of 15-20 minutes to less than two minutes using Build Cloud. This reduction in build time significantly improved their development cycle and allowed them to release updates faster.

**reduce build time to less than two minutes**

**E-Commerce Platform**

An e-commerce customer simplified their CI toolchain by leveraging Docker Build Cloud's dual AMD and ARM builders. Prior to Docker Build Cloud, they were using GitHub Actions, GitLab runners, and a custom GitLab runner to handle ARM architecture. Docker Build Cloud's native support for multi-architecture builds reduced their complexity and sped up their pipelines.

**reduced their complexity and sped up their pipelines**

These success stories illustrate how Docker Build Cloud's features deliver significant business value and return on investment for leading companies by optimizing build times, enhancing developer productivity, and enabling seamless multi-architecture support, ultimately driving profitability and competitive advantage.

# Docker Build Cloud is Available to Teams of All Sizes

Development teams of all sizes can leverage cloud-based build solutions to optimize their workflows and enhance productivity.

**Included Build Cloud Minutes:** Docker Pro, Docker Team, and Docker Business include Build Cloud minutes, allowing you to immediately experience the benefits. Use your included build minutes to evaluate how Build Cloud can accelerate your build times, streamline your development processes, and reduce overall costs.

**Scalable Usage:** Docker plans support adding additional minutes when your needs grow through a consumption-based model, making it a cost-effective solution for teams of all sizes.

**Support Resources:** Regardless of your plan, you will have access to Docker's robust support resources, including documentation and community forums to help you maximize the benefits of Build Cloud. Pro, Team, and Business plans also have access to commercial support.

We've designed Docker Build Cloud to provide a flexible, scalable solution that meets the needs of diverse development teams. The ability to offload builds to the cloud, utilize shared caches, and support multi-architecture builds ensures that your development processes are optimized for speed and reliability.

# How Docker Build Cloud and Docker Business Work Together

Out of all of the plans, a Docker Business subscription includes the highest level of Build Cloud minutes, Build Cloud cache, and builder sizes. Together, Docker Build Cloud and Docker Business ensure your developer teams have the best possible experience building and deploying software with reduced wait times, consistent build environments, and advanced management tools.

A major advantage of Docker Business is enhanced security.

We all know that code security has become a top-of-mind issue, but did you know that, according to Gartner, half of businesses have seen software-related security issues during the past six months?

Ask any of the thousands of North American car dealers who had to revert to pen and paper after the CDK Global vertical software stack was compromised. They know how vital it is for mission-critical software to be secured.

Docker Business is key to improving your company's security because it provides advanced management and visibility tools, especially designed for enterprises and large organizations:

**Security Features:**

- Hardened Docker Desktop: An extra layer of security by hardening container isolation to prevent malware from breaching the Docker Desktop Linux VM.
- Image Access Management: Developers get to control which container images they can access and use from Docker Hub, or other registries.
- Single sign-on (SSO) and SCIM: Supports SAML SSO and SCIM for centralized user management and access control.
- Docker Scout's vulnerability analysis and health scores help customers maintain security and compliance with ease.

**Management Capabilities:**

- Centralized Management Plane: Enables organization-wide management of all Docker development environments.
- Registry Access Management: Provides control over which container registries developers can access.

**Holistic Insights:**

- Docker Business provides visibility into how users are using Docker Desktop across the organization through dashboards and reporting, including version data, user data, Hub usage data, container usage data, as well as images and extensions used.

**Developer Productivity:**

- Docker Debug: Empower developers to produce high-quality applications by enabling earlier and easier debugging, enhancing confidence in their work.
- Developer Activation: Docker Init automates and simplifies the creation of Dockerfiles and Compose files, streamlining project setups and removing complexity.
- Cross-team Collaboration: We've fostered collaboration with investments like shared Docker Build Cloud builds in Docker Desktop, enhancing team interactions.

By offering these comprehensive features, Docker Business works with Docker Build Cloud to ensure that your development processes are not only efficient but also secure and compliant with industry standards.

# Transforming Development with Docker Build Cloud

Addressing inner loop slowdowns is crucial for improving developer productivity and satisfaction. Docker Build Cloud provides powerful solutions to these common challenges, enabling tech leaders to enhance their development processes with increased ROI.

## By reducing long and expensive build times, simplifying tool management, and improving collaboration, Docker Build Cloud is designed to help your dev team thrive.

With Docker Build Cloud, you will get the most from your team by helping them cut costs so they can focus their time and expertise on shipping better software while avoiding common pitfalls and frustrations.

Unlock your team's full potential and achieve faster, more reliable software delivery by building at build.docker.com.