

## Case Study

# How a Beauty Giant Achieved 25% Cost Savings with Container-First Development



**Top beauty company:** A top cosmetics retailer known for innovation in beauty and personal care products.

**Modernization effort:** Transitioned from monolithic infrastructure to modern, containerized architecture.

**Competitive edge:** Achieved faster deployments and cost efficiency, maintaining industry leadership.

## Highlights



### 60% faster deployments

Reduced average deployment times by 60%, enabling the company to rapidly push updates and new features to market, maintaining a competitive edge in the fast-paced beauty industry.



### 25% lower infrastructure costs

Through enhanced CPU and memory efficiency, Docker slashed infrastructure costs by 25%, freeing up resources for further innovation and growth.



### 1% deployment failure rate

Deployment failure rates were reduced from 8% to 1%, ensuring consistent, high-quality service delivery.

## Introduction

As a global leader in the beauty and personal care industry, this company's growth was feeling the impact of its outdated, monolithic infrastructure. With increasing digital demands, the company faced significant challenges – deployment cycles dragged on and inconsistencies across development environments led to delays and errors. These issues threatened the company's ability to innovate and maintain customer satisfaction and led to frustration among developers, risking the loss of top talent essential for driving future growth.

The company recognized the need for a modernized approach to address these challenges. Transitioning to a container-first development strategy with Docker, it aimed to streamline operations, reduce costs, and enhance scalability. Such a container-first approach prioritizes containerized applications from the outset, ensuring consistency and scalability across all development stages. This strategic shift resolved immediate operational inefficiencies and positioned the company for sustained growth and continued market leadership.



# Table of Contents

02	Challenge	05	Results and outcomes
03	Solution	06	Conclusion
04	Key benefits		

## Challenge

# Modernizing a legacy tech stack

As the company's digital operations expanded, its aging legacy infrastructure quickly became a bottleneck, leading to significant operational inefficiencies. The core issues stemmed from inconsistent development environments, monolithic applications, and cumbersome deployment processes that needed to be revised. These once-sufficient legacy systems had become outdated and were no longer capable of supporting the demands of a modern, fast-paced market.

The monolithic architecture also made scaling individual components challenging, resulting in frequent deployment delays and a high rate of errors. What should have taken hours stretched into days, leading to frustration within the team. The inflexibility of the legacy systems made it increasingly difficult to maintain the company's reputation for timely product rollouts and quick responses to market trends.

These challenges were not only technical but also personal for the team members. Before Docker, developers spent a lot of time getting their local setups to match production. Teams needed to be developing new features but instead lost valuable time troubleshooting. These ongoing frustrations took a toll on productivity, dampened team morale, and made it increasingly difficult to maintain a positive and productive work culture.

Scalability was another primary concern. As the number of applications and services grew, the existing infrastructure struggled to scale, compounding the deployment challenges. The system architecture for the monolithic applications was tightly connected, so even minor changes required redeployment of the entire application. This outdated process slowed updates and increased the chance of errors, putting the company at risk of falling behind competitors who could move more quickly and nimbly with a microservices approach.

The consequences of these technical challenges were far-reaching. Delayed deployments slowed innovation and threatened customer satisfaction, a critical factor in an industry driven by consumer expectations and rapid change. The company recognized that its ability to compete and grow in an increasingly digital market without addressing these legacy infrastructure issues was at serious risk.



## Solution

Recognizing the limitations of its monolithic infrastructure, the company transitioned to a microservices architecture as part of its modernization strategy. Containers were critical to this shift, and the company chose Docker for its reliability in local development. Docker enabled developers to build consistent, portable containers, ensuring that each microservice could be developed and tested independently. This approach allowed for quicker updates, better scalability, and more reliable performance, setting the stage for continued innovation and growth.

The implementation was carried out in phases, starting with non-critical applications to minimize risk and allow teams to learn and adapt. [Docker Engine](#) was used to containerize these applications, ensuring consistent environments that drastically reduced errors and deployment delays. This phased approach allowed the company to refine the process. It contributed to a renewed enthusiasm among developers, who could now focus on creative problem-solving rather than tedious manual setups. The team was motivated by the opportunity to embrace new technology with more collaborative practices. As teams gained confidence, [Docker Compose](#) was introduced to simplify the management of multi-service applications and helped devs work together more effectively across teams.

The experience of adding Docker to the CI/CD process was smooth, aided by Docker's [reliable documentation](#), established standards, and extensive integration partners. As the phased implementation progressed successfully, the company extended microservice architecture to more critical applications, ensuring its entire digital infrastructure benefited from the newfound efficiency. Now, teams deploy several times a day without worrying about stability.

Throughout this process, the company's development and operations teams collaborated closely, continuously refining their container-first approach to maximize the benefits of their new toolset. Engineering leaders encouraged shared ownership and promoted open communication. The result was a streamlined, scalable solution that addressed the core challenges and positioned the company to meet the demands of a rapidly evolving market.



## Key benefits



### Faster deployments with layer caching

#### [Docker's layer caching feature](#)

sped up builds, reducing deployment times by up to 67%. What used to take nearly an hour now only takes a few minutes. This faster process helps the team release updates quickly and work on more tasks, increasing productivity and saving money.



### Standardized environments with Docker Engine

By embracing a container-first approach, the company created consistent environments across development, testing, and production stages, significantly reducing deployment errors and leading to a smoother and more reliable process.



### 30% increase in developer satisfaction

The shift to modernized, containerized infrastructure significantly boosted developer morale, fostering a more collaborative work environment crucial for the company's ability to deliver high-quality products.



### Simplified management with Docker Compose

Docker Compose allowed for the orchestration of multi-container applications, streamlining the management of complex services and improving overall operational efficiency.



### Increased deployment frequency

The improvements in consistency and build speed allowed the company to increase its deployment frequency from bi-weekly to multiple times daily, enhancing its responsiveness to market demands.



### Optimized resource utilization

The company reduced CPU and memory usage by 50% through its container-first strategy. By deploying applications in Docker containers across several hosts, they could consolidate more instances onto each server. Containerization architecture reduced infrastructure costs by 25% and led to more efficient management of resources.



### Deployment failure reduced from 8% to 1%

The standardization and simplification provided by containerization reduced the deployment failure rate from 8% to 1%, ensuring greater reliability in the company's operations. With Docker, the testing environments are exactly like production, ensuring teams can be sure their tests are valid.



## Results and outcomes

The container-first development approach profoundly transformed the company's operational efficiency and overall business performance. One of the most significant improvements was in the deployment process, where average deployment time was reduced by 60%. This increase in speed enabled the company to push updates and new features to market much more rapidly, which was crucial for maintaining its market advantage in the fast-paced beauty and personal care industry.

### Enhanced scalability and cost efficiency

The company's scalability saw a substantial boost, with the number of Docker containers in operation expanding from 100 to more than 1,000. This growth ensured the infrastructure could comfortably handle increasing demand without sacrificing performance. In addition to enhanced scalability, the company achieved a 25% reduction in infrastructure costs, driven by a 50% increase in CPU and memory efficiency. These savings allowed for more effective resource allocation, supporting ongoing innovation and operational improvements.

### Improved stability and reliability

The company saw a significant boost in the reliability of its operations, with deployment failure rates dropping from 8% to just 1%. Alongside this improvement, system uptime increased from 99% to 99.8%, significantly reducing downtime and ensuring that services were consistently available to customers. These enhancements minimized disruptions and helped the company maintain its reputation for delivering reliable, high-quality services. The combination of fewer errors and better service availability was crucial in a competitive market where reliability is critical to success.

### Business and cultural impact

Beyond these quantitative improvements, adopting Docker had a broader impact on the company's strategic objectives. The accelerated deployment times and enhanced scalability enabled the company to maintain its leadership position, ensuring it could more effectively meet customer expectations and outpace competitors. The cost efficiencies gained from optimized resource utilization provided greater financial flexibility, which the organization could reinvest in future growth and innovation initiatives.

Furthermore, the cultural benefits of implementing more modern software practices were significant. Developer satisfaction increased by 30%, reflecting the smoother, more efficient processes. This boost in morale fostered a more collaborative and productive work environment, driving further innovation and enhancing the overall work culture within the development teams.



## Conclusion

The transition to a container-first development approach, executed in multiple phases, has driven substantial improvements across the company's development and operations. By cutting deployment times by 60%, enhancing scalability, and reducing infrastructure costs by 25%, teams effectively resolved the operational challenges threatening the company's ability to compete. These results addressed immediate issues and set the stage for long-term success in a fast-moving market.

The improvements in scalability and the significant reduction in deployment errors have fortified the company's infrastructure, ensuring it can confidently meet growing customer demands. Additionally, a 30% increase in developer satisfaction reflects the positive cultural shift within the organization, fostering a more innovative and efficient work environment.

Adopting Docker wasn't just about solving the current problems; it was about setting up operations for the future. The organization sees the forward-looking benefits of Docker's implementation as a step in line with its broader growth and innovation strategy to meet market demands.

These outcomes create a solid foundation for continued growth and innovation. With Docker now integral to its operations, the company is better equipped to navigate the complexities of an increasingly digital landscape. This success ensures the company can maintain its leadership in the beauty and personal care industry, driving further innovation and responding to market demands with enhanced agility.

Embracing Docker within a container-first development strategy has addressed the company's immediate operational needs and paved the way for sustained success, ensuring the company remains a competitive force in the industry.

### Learn more

Subscribe to the [Docker Newsletter](#).

New to Docker? [Get started](#).

Get the latest release of [Docker Desktop](#).

Vote on what's next! Check out our [public roadmap](#).

Have questions? The [Docker community is here to help](#).

*This case study was contributed by [Vladimir Mikhalev](#), also known as Valdemar, a DevOps Consultant and Team Lead at Ataccama. With more than 20 years of experience at companies like IBM and Amazon, Vladimir's expertise in Docker and containerization shines through. His [practical insights](#) offer a clear path for organizations to optimize their operations with modern IT practices.*

