

Case Study

How Kapa.ai Achieved Super-Fast, Production-Ready Local Environments with Docker



About Kapa.ai: Kapa.ai enables developer-facing companies to create AI-powered support and onboarding bots, enhancing developer experience and reducing support efforts.

Industry: AI-powered developer tools

Location: San Francisco, CA

Kapa.ai needs local development because of its large machine requirements and preference for developer autonomy. Docker gives them flexibility and reduces their costs.

Highlights



\$50k per year saved. Docker allows Kapa.ai to run a full clone of the production system locally. Using Docker means effective testing and development while reducing the costs of provisioning resources separately.



Flexibility and speed. Docker supports local development, making setup and testing quicker and easier. This makes it easier for developers to iterate and experiment without the cumbersome process of managing cloud resources.

"Docker just works so well. It lets us quickly set up and experiment with new components. Without it, provisioning these services would be much slower and way more complex."

Finn Bauer
CTO at Kapa.ai



Table of Contents

02	Introduction	05	Solution
03	Identified opportunities	05	Implementation process
03	Use of Docker	06	Key benefits
04	Challenges	07	Results and outcomes
04	Business challenges	07	Conclusion

Introduction

[Kapa.ai](#) is a leading AI company specializing in enhancing search functionalities within extensive documentation and knowledge bases. Faced with the challenges of managing and making best use of large volumes of complex data, Kapa.ai uses advanced AI to streamline and optimize search processes. They make it easier than ever for users to quickly and accurately find the information they need.

In their day-to-day operations, Kapa.ai must ensure that their local development environments precisely replicate their production systems. This replication is crucial for accurate testing and development, but it can be both resource-intensive and time-consuming. To meet these challenges, Kapa.ai has integrated Docker into their workflow.

Docker's containerization technology allows Kapa.ai to create consistent, reliable local environments that precisely mirror their production setup. Using Docker means their developers can test and iterate quickly, saving hundreds of hours of effort and stress.

Reducing setup times and improving overall efficiency offer Kapa.ai clear benefits. By using Docker, Kapa.ai not only addresses immediate operational challenges but also positions themselves for growth by enhancing their development capabilities and accelerating their test and improvement cycles.



Identified opportunities

Kapa.ai faced major challenges with local development and testing, primarily due to the resource-intensive nature of machine learning (ML) development. As the company's AI models grew more complex and demanding, the need for serious computing power became more and more apparent.

This posed a challenge for Kapa.ai, because their existing cloud and hybrid compute solutions started to show limitations when scaling their AI/ML development. Processing such large amounts of data and running complex algorithms required powerful machine capabilities.

Kapa.ai also needed to ensure that their production environments were replicated locally for effective testing and development. This replication was critical to maintaining consistency and accuracy across the different stages of their workflow. Kapa.ai wanted to guarantee both the quality of their tests and make the workflow as stress-free for their developers as possible.

Kapa.ai wanted to streamline their local development processes without compromising on performance or reliability. They needed a tool that could handle their intensive compute requirements while also providing the flexibility to iterate and experiment – fast. However, once Kapa.ai started scaling their AI and ML development, they realized their computing requirements were much bigger than originally forecasted, and their cloud and hybrid solutions just couldn't keep up.

This is where Docker came into play, offering the capabilities to meet these demands and to support Kapa.ai's growth in AI/ML development.

Use of Docker

Docker enabled Kapa.ai to run a full clone of their production system locally

Kapa.ai faced several challenges in replicating their production system for local development and testing. Managing cloud resources was cumbersome and time-consuming, often involving finding a provider, paying for services, and setting up instances that the whole team could access. It was a multi-step process that slowed iteration, added blockages to their dev workflows, created bottlenecks, and increased costs.

Docker provided the solution. By enabling Kapa.ai to run a full clone of their production system locally, Docker allowed the team to replicate and test everything quickly, easily, and affordably. With Docker, they could set up isolated environments in minutes, ensuring consistency and accuracy across different stages of development.

Docker eliminated the need for extensive resource provisioning, saving Kapa.ai significant costs as well as headaches. As a result, the team gained the flexibility to experiment and improve without the need to oversee multiple different instances and sharded database clusters.

Finn Bauer, CTO at Kapa.ai, sees these effects across the team. "Docker's impact on our speed of iteration has been huge. When we use cloud services we have to find a provider, pay for it, somehow figure out how to share the instance or even have multiple instances set up that the whole team can access. It can take so long and have so many steps.

With Docker, you get a service in Docker compose and it's all done in literally 10 minutes. It makes everything super easy to scale and it means we don't have to pay for any extra infrastructure. This also probably saves us in the magnitude of \$50,000 a year."



Challenges

Facing challenges in local development and testing large models

Kapa.ai faced several technical challenges in their local development and testing processes. Replicating their production environments locally proved difficult, often because of the resource-intensive nature of AI-specific features and large models.

Making sure their test environments were consistent and reliable was critical for accurate testing, but achieving this was too often complex and time-consuming. This need for a reliable environment often clashed with the limitations of their existing cloud and hybrid solutions.

A key pain point was the inconvenience of dealing with cloud infrastructure. As Bauer explains, “I don't want to think about how to share the database cluster between people. I like the experience of using Docker so much more. If I want to change the version for trying something, I don't want to reach out to the cloud support to change the version, I just want to change locally. Docker lets me do that.”

Docker addressed Kapa.ai's challenges by giving them a consistent, reliable environment for local development and testing. Using Docker let Kapa.ai focus on improvement without being bogged down by the complexities of cloud infrastructure management.

Business challenges

Kapa.ai also faced business challenges that hindered their development processes and impacted their overall efficiency. One major issue was the high costs associated with provisioning separate resources for each developer. Maintaining multiple instances and ensuring that each developer had the necessary resources was not only expensive but also resource-intensive. This financial burden constrained the company's ability to allocate funds to other critical areas of the business, slowing their growth.

More than just high costs, Kapa.ai were burdened with inefficiencies in their development workflow because of the inconsistent cloud/hybrid environments. Too often, their developers encountered discrepancies between their local development setups and the production environment, which led to errors and delays.

These inconsistencies were annoying and stressful, and required additional time and effort to resolve.

The inefficiencies Kapa.ai were facing contributed to delays in development and testing cycles, which had a direct impact on project timelines. Understandably, this caused frustration and created unwanted friction in the team's workflows.

Addressing these issues was a priority for Kapa.ai so they could improve their development processes, reduce costs, and enhance their ability to deliver the premium-quality AI solutions they had become known for.



Solution

Docker makes local testing quick and easy

Kapa.ai used Docker to make their local development and testing quick, easy, and accurate — every time. Using Docker's containerization technology meant Kapa.ai could create full clones of their production system locally, immediately.

By creating isolated environments quickly and efficiently, the team iterated faster and tested new features with more ease, fewer delays, and less stress. This approach both improved their development speed and reduced the costs associated with provisioning separate resources for each developer.

Bauer goes on to say: "We've been using Docker from the start. Our cloud service is Google and in total we've done two big service and database merges. We did a load of dry runs in Docker before production, and it made the merge so simple. The transition was really smooth."

Using Docker for dry runs before production meant Kapa.ai could identify and resolve potential issues early, leading to smoother transitions and more reliable deployments. This approach helped minimize risks and amplified their ability to deliver high-quality AI solutions.

Implementation process

Kapa.ai set up Docker containers to mirror their production environment on each developer's laptop. The containerization approach allowed the team to run a complete system locally, giving them vital accuracy in replication and testing.

Bauer explains: "Recently, we tested out a new RabbitMQ messaging service and we could do it so quickly and start gathering real data on how it impacted our product and everything because it was so fast and easy to do with Docker. It would have been a huge hassle to test our RabbitMQ without Docker, it would have taken so much longer and would have been way more complicated. Docker just saves us so many headaches and so much money — replicating production configs is way more expensive than using Docker."



Key benefits

Improved development speed, cost savings, and more flexibility



Improved development speed

Using Docker means Kapa.ai can set up and test new components quickly. With Docker, local development becomes faster, which means the team can iterate and experiment at a much faster pace. Developers can bypass the cumbersome process of managing cloud resources, which traditionally drags down speed. This rapid setup and testing capability also enhances the development cycle, allowing Kapa.ai to bring new features and improvements to their AI models more efficiently.



Cost savings

Using Docker for local environments reduces the need for separate resource provisioning, minimizing the costs associated with replicating production systems. By avoiding the dependency on costly cloud infrastructure for local testing, Kapa.ai saves an estimated \$50,000 per year.



Better flexibility

Docker provides the development team with the flexibility to experiment with new setups easily. The local development capability gives developers the autonomy to work quickly without relying on external resources. It creates a much better development experience and accelerates testing and improvement processes.

As a Bauer says, “when I think of Docker, it’s about the value, the flexibility you have locally, and how easy, how good local development is!”

Docker’s impact on Kapa.ai extends beyond technical benefits, fostering a more agile and responsive development environment that supports ongoing improvements and helps Kapa.ai keep their competitive advantage.



Results and outcomes

Kapa.ai achieves faster iterations and experimentation



Docker reduced the time required to set up development environments.



Developers report much quicker turnaround times for testing new features.



Kapa.ai avoided the high costs associated with extensive cloud resource provisioning.

The team enjoyed increased efficiency in their testing and development cycles, allowing for quicker iterations and faster deployment of new features. Positive feedback from the developers highlighted the newfound ability to experiment more freely, without the constraints of cost concerns or resource limitations. Again, a major productivity boost. Increased ease and efficiency of testing also led to a number of downstream benefits for team morale and energy.

Conclusion

Enabling quick setup and testing for future projects

By adopting Docker, Kapa.ai overcame the major challenges of replicating production environments, managing cloud resources, and maintaining consistency across development stages. Docker enabled Kapa.ai to streamline their workflow, save \$50,000 and level-up the speed and accuracy of their development processes.

"It's the best local development solution. Having a full clone of our production systems means we can test and develop really quickly with zero fuss. It saves us tons of time because we don't have to do all the slow and cumbersome processes involved in connecting to a cloud database and managing all of that every time we want to test," Bauer says.

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](#).

