# Modernizing AI Workflows: How Containers Make It Possible

By: Yiwen Xu

# Introduction

The rapid advancement of Generative AI (GenAI) and agents is transforming businesses across industries. Every company, whether in finance, healthcare, retail, or technology, is becoming an AI company. Companies adopting this new technology are seeing tangible results from their GenAI investments, with increased revenue and reduced costs. A Google Cloud ROI study found that 74% of enterprises achieve ROI within the first year of GenAI deployment, and nearly half (45%) report at least doubled employee productivity.

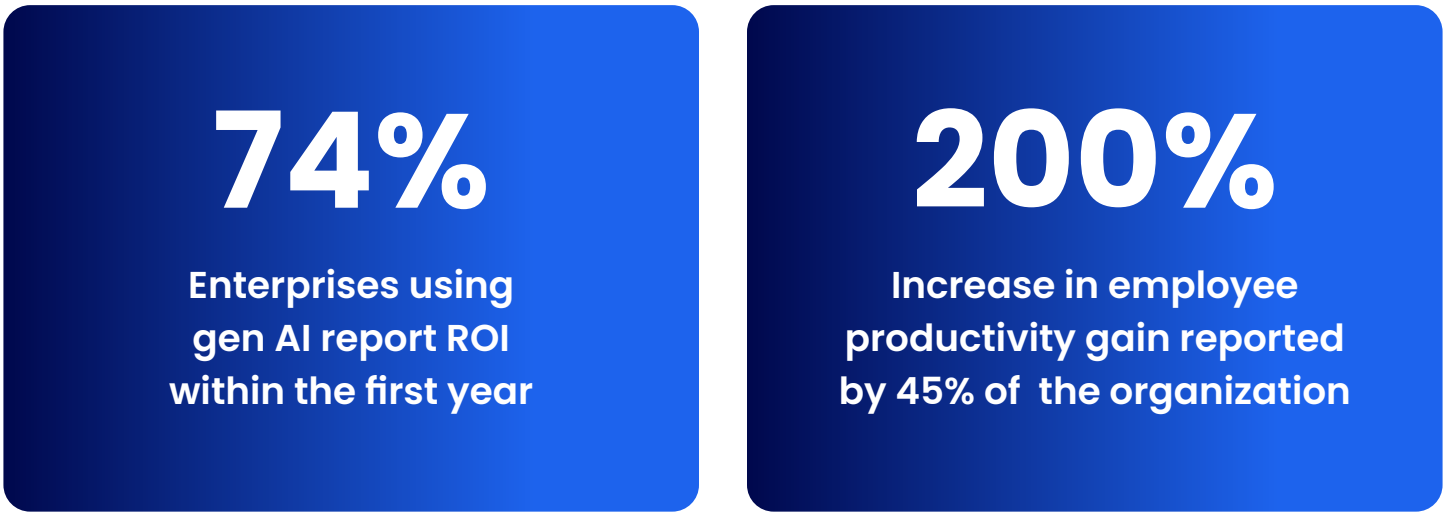



**Figure 1:** AI is more than just hype, with companies seeing tangible results per a Google Cloud Study.

As AI adoption accelerates, organizations are exploring various use cases and how AI can fuel their growth. But the landscape is evolving quickly. New models, improved performance, and emerging technologies are reshaping the field. DeepSeek is a clear example of how fast the AI landscape is evolving. In just a short time, it's delivered high-performing open models that rival proprietary systems. Its strong performance benchmarks show that innovation in model architecture and training efficiency isn't slowing down; it's accelerating. Tools and agents built just months ago can already feel outdated as new models like DeepSeek reshape what's possible in real time.

With change happening at such a rapid pace, staying competitive requires speed, the right technology, and strong strategic partnerships. In this whitepaper, we'll break down the key technology foundations for AI adoption, focusing on why containers are essential for building and scaling AI applications. Finally, we'll explore how the right partner can empower organizations to develop GenAI solutions efficiently, securely, and at scale.

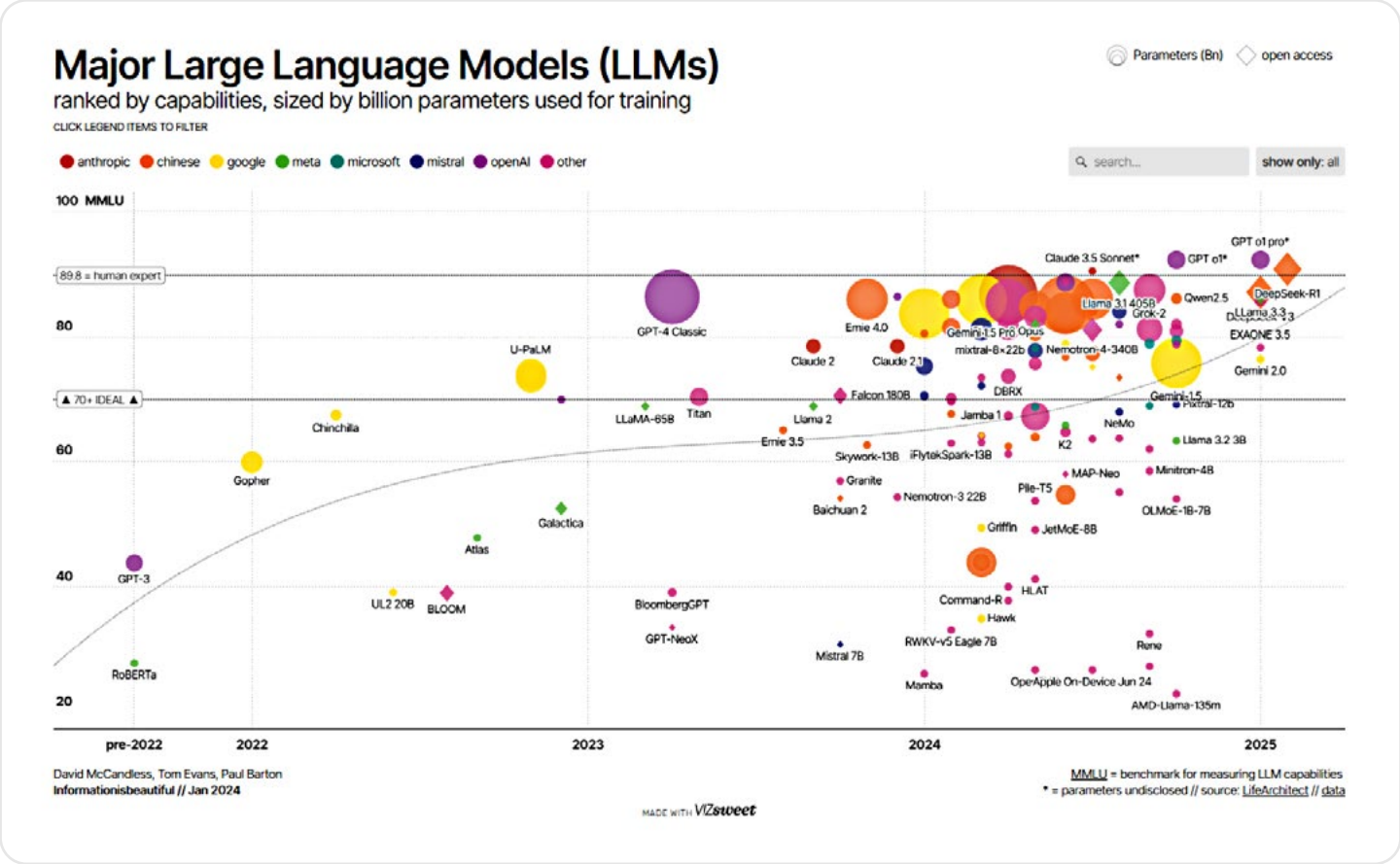



**Figure 2:** The AI landscape is evolving quickly with new models and advancements coming out at unprecedented speed. (Credit: informationisbeautiful)

## Chapter 1

# The right technology: Why containers are the backbone for AI development

- Cloud-native transformation is step zero — monoliths can't keep up with AI's pace.
- Containers offer consistent, scalable, secure environments to deploy GenAI apps.
- With containers, teams can reuse existing DevOps practices for AI without starting from scratch.

## Cloud-native first, then AI

Before companies can fully embrace AI, they must first undergo a cloud-native transformation from monolithic architecture.

Monolithic systems, while once effective, fail to meet the demands of today's fast-paced business environment. Their tightly coupled architecture makes it difficult to scale individual components, adapt quickly to changing customer expectations, or integrate emerging technologies like AI. Any change, no matter how small, often requires redeploying the entire application, leading to slower development cycles, increased risk, and higher operational costs.

In an AI world where agility, resilience, and innovation are critical, monolithic architectures become a bottleneck, preventing organizations from responding to market shifts, experimenting at speed, and delivering modern digital and AI experiences that drive growth.

On the other hand, cloud-native platforms are designed with modern application development in mind. Beyond speed and agility, cloud-native architectures also drive cost optimization and operational resilience. With built-in scalability and dynamic resource allocation, businesses can reduce downtime, improve system reliability, and better manage infrastructure costs.

Cloud-native development isn't just about moving to the cloud — it's about leveraging scalability, flexibility, and resilience to stay competitive. This approach allows teams to build, test, and deploy applications faster, with streamlined workflows from the start. Without this foundational shift, AI initiatives often stall or fail to scale beyond prototypes. Organizations that make this transition not only future-proof their operations but also position themselves to lead in innovation, customer experience, and long-term value creation.

Containers, microservices, and cloud scalability form the foundation of cloud-native development. They're also essential for running and managing AI workloads efficiently. Once you're ready to modernize development with cloud-native workflow, it's easy to shift to GenAI application development, which also leverages containers as the core foundation. Containers make using AI developer tools more productive. With smaller, modular microservices, AI systems can more easily understand the structure and context of your code.

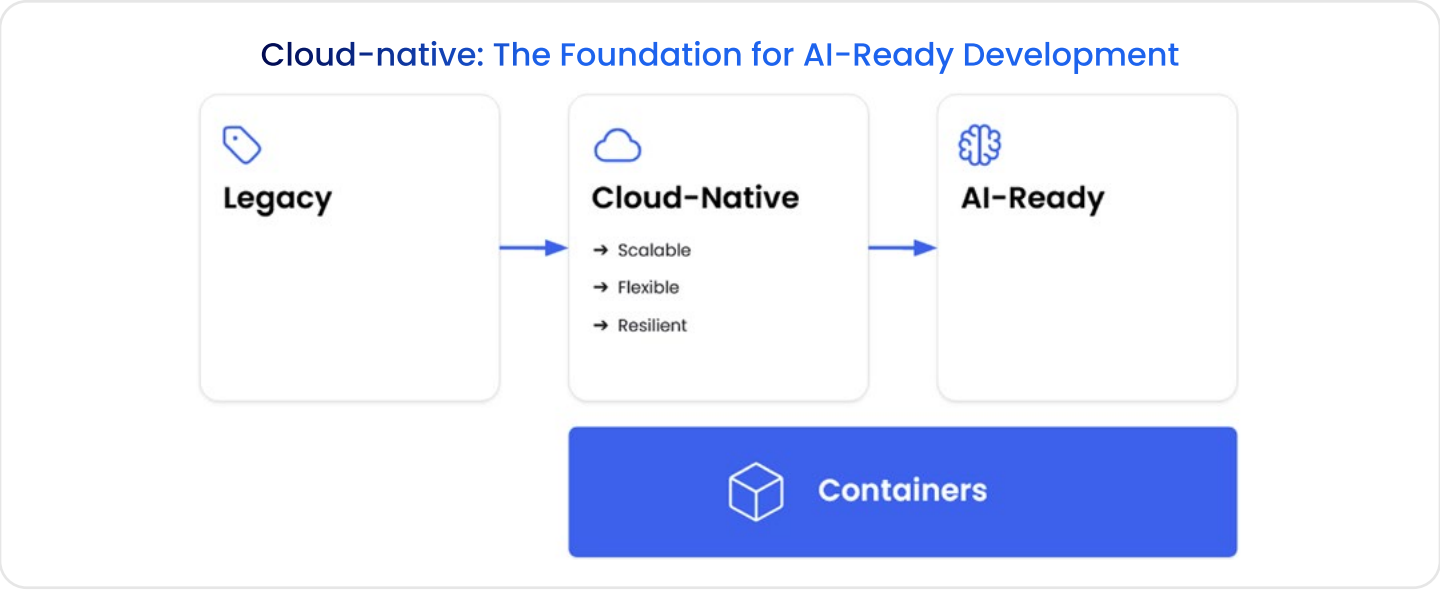



**Figure 3:** Companies must first undergo a cloud-native transformation from monolithic to be AI-ready

## What are containers?

Containers are standardized software units that package applications and their dependencies together, ensuring consistent performance across different environments and platforms. Unlike traditional virtual machines, containers share the host system's OS kernel but run in isolated user spaces. This makes them fast to start, efficient to run, and easy to scale. Whether you're building a microservice or deploying a full-stack app, containers make it easier to manage consistency across development, testing, and production environments. Think of them like shipping containers - they keep software components intact, secure, and portable.

Why does this matter? Containers give development teams the freedom to innovate with new technology without worrying about complex environment configurations. The result: faster, more reliable software delivery.

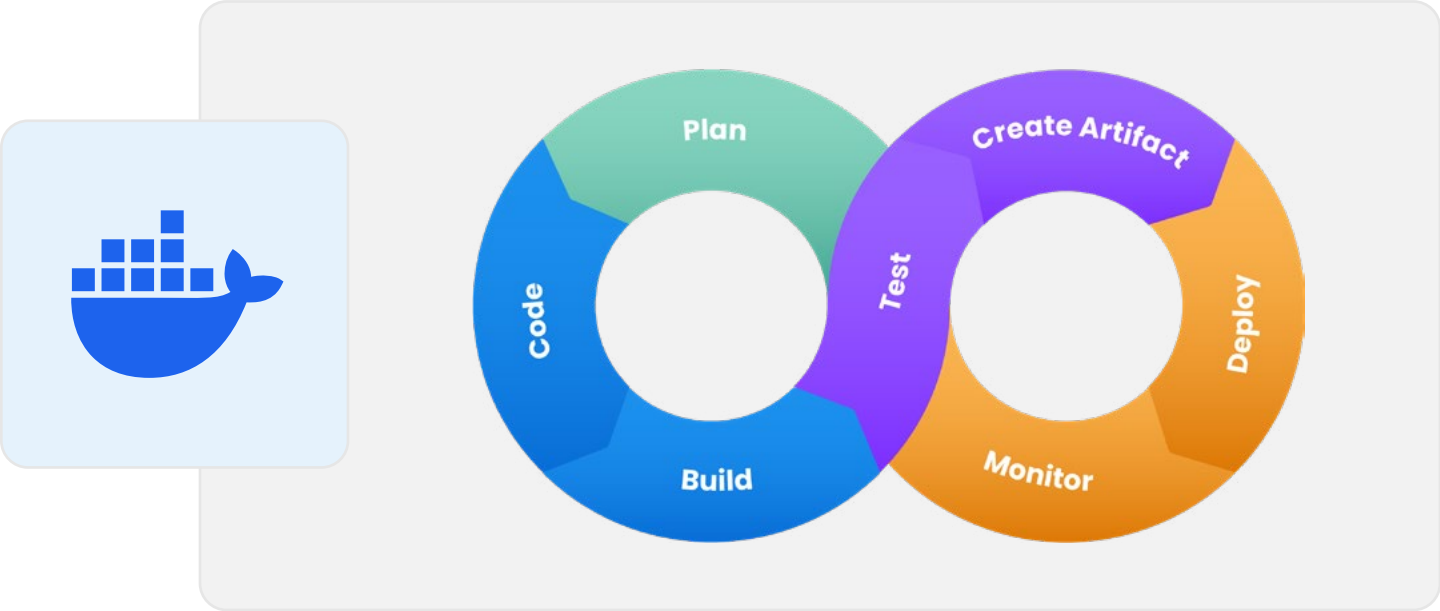



**Figure 4:** Containers enable more efficient and reliable delivery of apps

## Why containers for GenAI development?

### Efficiency

Easily package complex AI setups and iterate faster.

### Scalability

Quickly adapt to new models and tools without workflow disruption.

### Security

Container isolation reduces risk from AI dependencies and custom environments.

A Generative AI (GenAI) application is software that leverages large language models or other AI systems to generate content, automate tasks, or interact intelligently with users. GenAI applications may feel like a new challenge, but at their core, they're just another workload that fits naturally into existing cloud-native development processes. Containers provide a solid foundation for building, deploying, and scaling GenAI applications, offering three key advantages:

01

### Efficiency and seamless integration

Containers are a proven, efficient technology that accelerates modern application development. For example, one global beauty giant saw a 60% faster deployment after adopting containers. Since GenAI applications follow the same software development process as other workloads, teams can:

- Easily add AI capabilities to existing applications.
- Start new GenAI projects without redesigning their development infrastructure.
- Abstract AI complexity by packaging components and architectures (like RAG) into containers, streamlining development.

02

### Scalability and future-proofing

The AI landscape evolves quickly, but containers provide the flexibility to adapt. Over the past decade, containers have powered multiple tech innovations and continue to support the fast-changing GenAI space. Whether introducing a new AI component, developers can simply package it in a container without overhauling workflows or retraining teams.

03

### Security and isolation

Containers offer built-in security by isolating applications from their dependencies, each other, and the host operating system. This ensures that AI workloads run reliably and securely, reducing risks like dependency conflicts or hardware constraints.

### Enterprise adoption: The shift to containerized AI workloads

Industry trends show strong momentum toward containerized AI applications. A recent CIODive article noted that seven in ten enterprises plan to containerize their GenAI workloads, and that Gartner predicts that by 2027, over 75% of AI deployments will use containers, up from 50% today..

With cloud-native development on the rise, containers have become the default technology for AI applications. They provide the efficiency, scalability, and security that companies need to succeed in AI development now and in the future.


**Figure 5:** Enterprises are adopting containers for their GenAI applications.

## Chapter 2

# Addressing AI challenges with a container-based workflow

**With containers as the foundation, focus on these strategies to enable fast, secure AI development and boost productivity with intelligent agents:**

- **Empower experimentation** with flexible dev environments, trusted content, and tools that support rapid iteration and seamless cloud scaling.
- **Embed security throughout** — from verified content and isolated runtimes to policy-driven guardrails and compliance tooling.
- **Leverage workflow-native AI agents** customized with enterprise context and enhanced through Model Context Protocol (MCP) for greater relevance and reliability.
- **Think beyond code completion** — orchestrate multiple specialized agents to optimize the entire software development lifecycle.

Adopting containers is a great first step in integrating AI into your applications. The faster you can refactor legacy code into containers, the easier it becomes to infuse AI without disrupting existing workflows. Instead of a major overhaul, AI adoption becomes a process of continuous improvement and allows teams to pivot and scale as strategies evolve.

**As companies begin their journey with containers and AI, they often ask three key questions**:

1. How can developers easily build generative AI applications?
2. How can applications be secure and compliant from day one?
3. How can AI support developer productivity with intelligent agents?

Let's go through these challenges one by one and provide recommendations on how to overcome them.

## Empowering developers to build GenAI apps easily

Developers face several common challenges when working with GenAI, including skill gaps, hardware limitations, and quality assurance concerns. Companies can provide a structured, scalable development environment to address these issues.

### Bridging the Skills Gap

Many developers are still new to GenAI. According to an IBM study, 76% of enterprise application developers don't consider themselves experts in GenAI.


**Figure 6:** According to an IBM study, the majority of enterprise application developers don't consider themselves proficient in GenAI.

**To close this gap, organizations should provide:**

- A developer-friendly platform where teams can discover, explore AI models, orchestration tools, databases, MCP servers, etc.
- Quick-start resources like templates, sample applications, and scaffolding tools for faster prototyping.
- Local environments that allow developers to pull and run models with GPU acceleration and standard APIs for seamless integration.

By lowering the entry barrier, teams can experiment, iterate, and accelerate time to market.

## Overcoming Hardware Constraints

GenAI workloads demand significant computing power, often exceeding the capabilities of standard hardware.

**To address this, organizations should provide a flexible development environment that supports:**

- Local testing and experimentation for smaller workloads
- Leveraging available GPU resources on local machines when working with big models or accelerating inference
- Seamless cloud scaling for larger AI/ML models, i.e., GPU on demand

This hybrid approach ensures developers can work efficiently without being bottlenecked by hardware limitations.

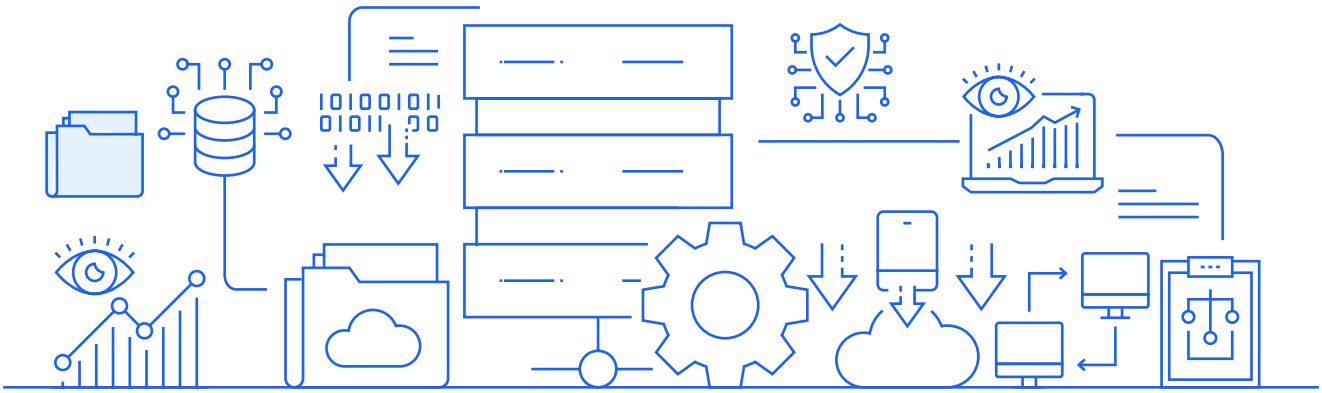
## Ensuring quality in AI applications

Unlike traditional software, testing GenAI applications is more complex. Large language models (LLMs) are inherently non-deterministic, meaning the same input can yield different outputs across runs, even with identical prompts. While non-determinism can be useful for generating diverse responses, it introduces uncertainty and risk in production environments. For teams building AI-powered applications, this behavior complicates testing, debugging, and reproducibility. Without guardrails, slight model fluctuations can lead to unexpected or even unsafe outputs, especially in high-stakes scenarios like customer support or content moderation.

**To improve reliability:**

- Early integration testing should be prioritized. Shift-left testing is a good practice that significantly improves defect detection rate and time and reduces context switching for developers.
- Provide tooling that offers reproducible methods to validate application logic that utilizes AI models. Ideally, the tools have preconfigured implementations for the most popular technologies such as databases, message brokers, cloud services like AWS, or other microservices for quick and easy test setup. For example, spinning up containers to mock and simulate service-level interactions.
- Scalable and cost-efficient testing tools that work locally and can be integrated into CI/CD pipelines.

With robust testing, developers can ensure AI applications behave predictably, minimize the margin of error, and meet quality standards.



## Building secure and compliant GenAI apps from day one

Now that the dev team has jump-started GenAI app development, security and compliance become top priorities. The IBM study also found that governance and compliance rank among the top five challenges in enterprise GenAI adoption - a valid concern, as AI introduces new security risks. We recommend approaching security with a layered approach, starting with a solid foundation and secure environment coupled with continuous monitoring and enterprise control.

### Laying a secure foundation

**GenAI expands the attack surface with additional components like LLMs and RAG architectures, increasing the risk of vulnerabilities. Organizations should start with:**

- Models, MCP tools, and frameworks etc. from trusted sources. Using verified, minimal base images from trusted registries helps reduce the attack surface and prevents vulnerabilities from creeping in via bloated dependencies or outdated packages.
- As AI tooling proliferates, so does the risk of unvetted components slipping into production. Guardrails for developer environments, such as access control tools, come in handy to enforce the use of approved AI components.

### Providing a secure environment and continuous monitoring

**Containers are secure by design, but proper configuration and additional safeguards are necessary, especially in regulated industries like banking and healthcare. Best practices include:**

- **Enhanced container isolation for stricter security**
  - For AI/ML environments where sensitive data or proprietary models are involved, these enhanced isolation layers help improve security posture.
- **Vulnerability analysis tools for continuous monitoring**
  - Continuous monitoring ensures that your images stay compliant and secure throughout the SDLC. For teams building AI workloads, where dependencies like CUDA libraries or Python packages are often updated, this visibility is key to avoiding silent regressions or shipping known vulnerabilities.
- **Enterprise control tools for governance and compliance oversight**
  - Enterprise platforms offer centralized control over who can push and pull images, enforce usage of approved registries, and define policies for software composition.

By maintaining full visibility and governance over the development process, organizations can mitigate risks while ensuring regulatory compliance.


**Figure 7:** A layered approach to security and compliance of GenAI applications.

## Supercharge developer productivity with AI agents

AI agents can significantly boost developer productivity but only when they are specialized, accessible, and customizable. To fully harness the power of AI, these agents should seamlessly integrate into existing workflows and align with your company's needs and policies.

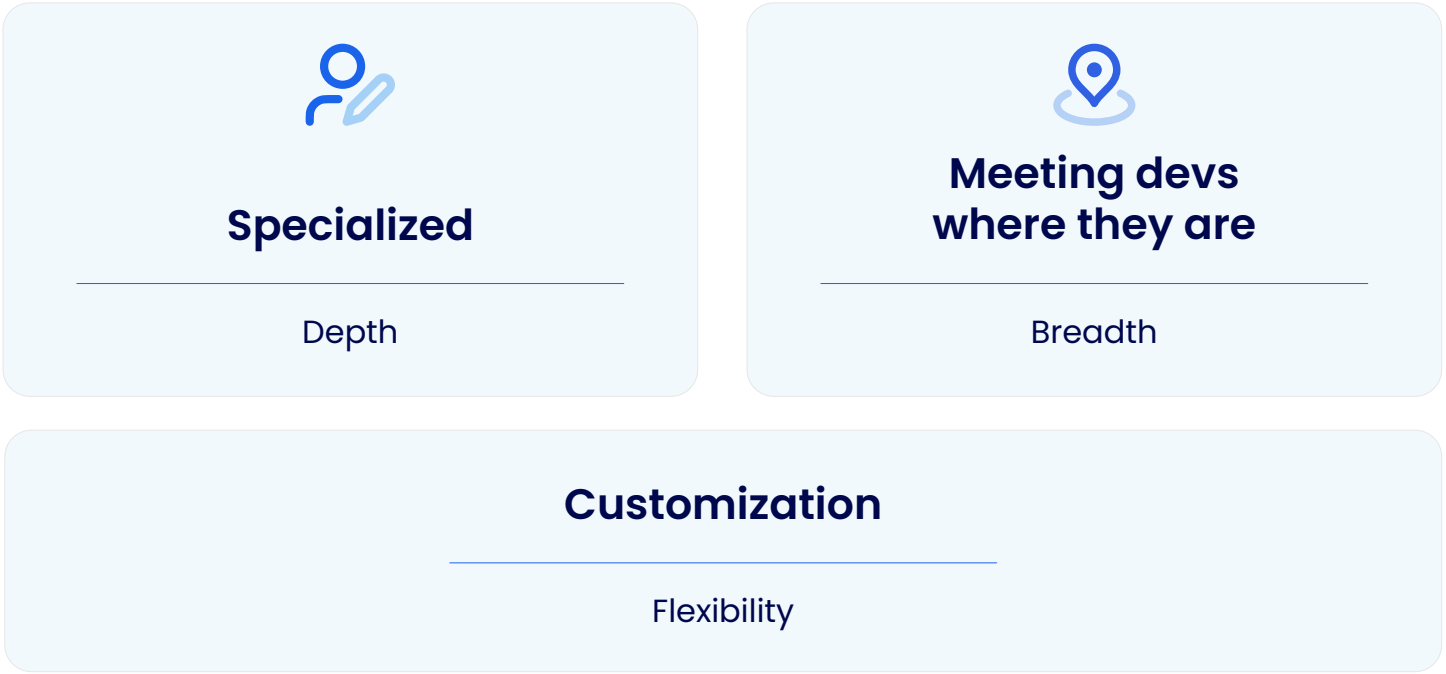



**Figure 8:** Agents are most useful when they are specialized, accessible, and customizable.

### Specialized AI Agents Are the Most Effective

AI agents are most useful when they are trained for specific tasks, such as performing container-related tasks. By learning from past usage patterns and user intent, these agents can provide actionable guidance and automation.

**To enhance their effectiveness, AI agents should have:**

- Access to historical usage data to recognize patterns and predict developer needs.
- Expert knowledge to help solve developers' problems and inquiries.
- Ongoing updates to stay relevant as technology evolves.

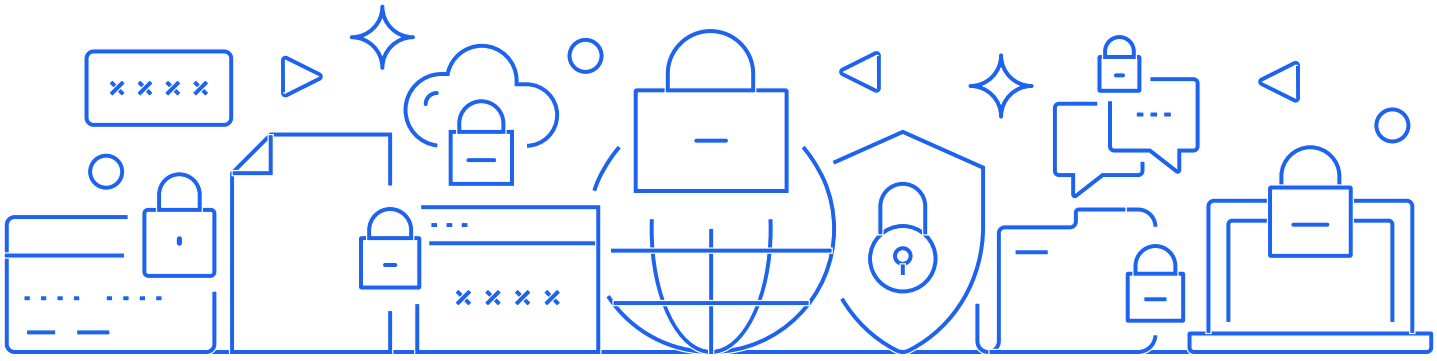
### Meet developers where they work

Constantly switching between tools disrupts productivity. That's why AI agents should be available where developers already work, whether that's on the desktop, command-line interface (CLI), or integrated development environments (IDEs).

By embedding AI directly into familiar tools, companies can drive higher adoption and engagement. This makes AI assistance a natural part of the development process.

### Customization makes AI agents more powerful

AI agents become significantly more valuable when they're tailored to specific enterprise needs. There are two key ways to customize AI agents. Model Context Protocol can be used to expand agent capabilities. And integrating enterprise-specific data and infrastructure makes agents more accurate and secure.



### Expanding AI agent capabilities with Model Context Protocol

Model Context Protocol (MCP) is an open-source standard from Anthropic that allows AI agents and LLMs to connect with external data sources and tools. With MCP, AI Agents can retrieve data from external sources, perform operations with third-party services, or even interact with local filesystems.

MCP works by introducing the concept of MCP clients and MCP Servers. This means that clients request resources and the servers handle the request and perform the requested action. MCP Clients are often embedded into LLM-based applications, such as the Claude Desktop App. Then, the MCP Servers are launched by the client to perform the desired work using any required additional tools, languages, or processes. For example, an AI agent specializing in containers can:

- Create a project on GitHub using the MCP GitHub server.
- Access the internet via the MCP Fetch server.

And since containers are ideal for running MCP servers, setup is effortless. Just spin up a container, and you're ready to go. Containers aren't just about packaging, they give us a controlled runtime environment where we can add guardrails and build a safer path toward adopting MCP servers.

### Customizing AI agents with enterprise context

MCP is still in its early stages, so most enterprises will likely focus on customization through their own context. By integrating enterprise data, internal policies, and compliance requirements, companies can make AI agents more relevant and reliable for their teams and ensure AI responses align with company-specific best practices. Additionally, organizations can use custom AI infrastructure and preferred LLMs to power AI agents.

### Beyond coding agents: Optimizing the entire SDLC with AI

RedMonk analyzed Google's DORA Report, which revealed a surprising insight: while AI-assisted workflows improve individual productivity, increasing AI usage by 25% reduces overall throughput, stability, and time spent on valuable work. One possible reason? AI is improving only one part of the software development lifecycle (SDLC), while bottlenecks elsewhere slow everything down.

To truly enhance SDLC performance, companies must identify constraints and remove barriers - not just speed up coding.

Most AI agents today focus on coding assistance, making software development faster. But optimizing just one part of the SDLC isn't enough. Instead, AI should enhance productivity across the entire development process, from planning to deployment.

To solve this, we can implement a multi-agent system where different AI agents handle specific tasks throughout the SDLC, collectively improving team efficiency.

### AI agents as workflow orchestrators

As AI agents become more advanced, they won't just assist developers. They'll also become more like a coworker, collaborate across teams, automate workflows, and drive business outcomes.

Imagine an AI-powered environment where one AI agent orchestrates multiple specialized agents, automating tasks across multiple cross-functional teams. Agents can handle tasks like merging code, distilling PRDs, designing UI, writing Jira tickets, and suggesting proactive improvements. This shifted workload means teams can spend less time on repetitive tasks and more time on innovation.

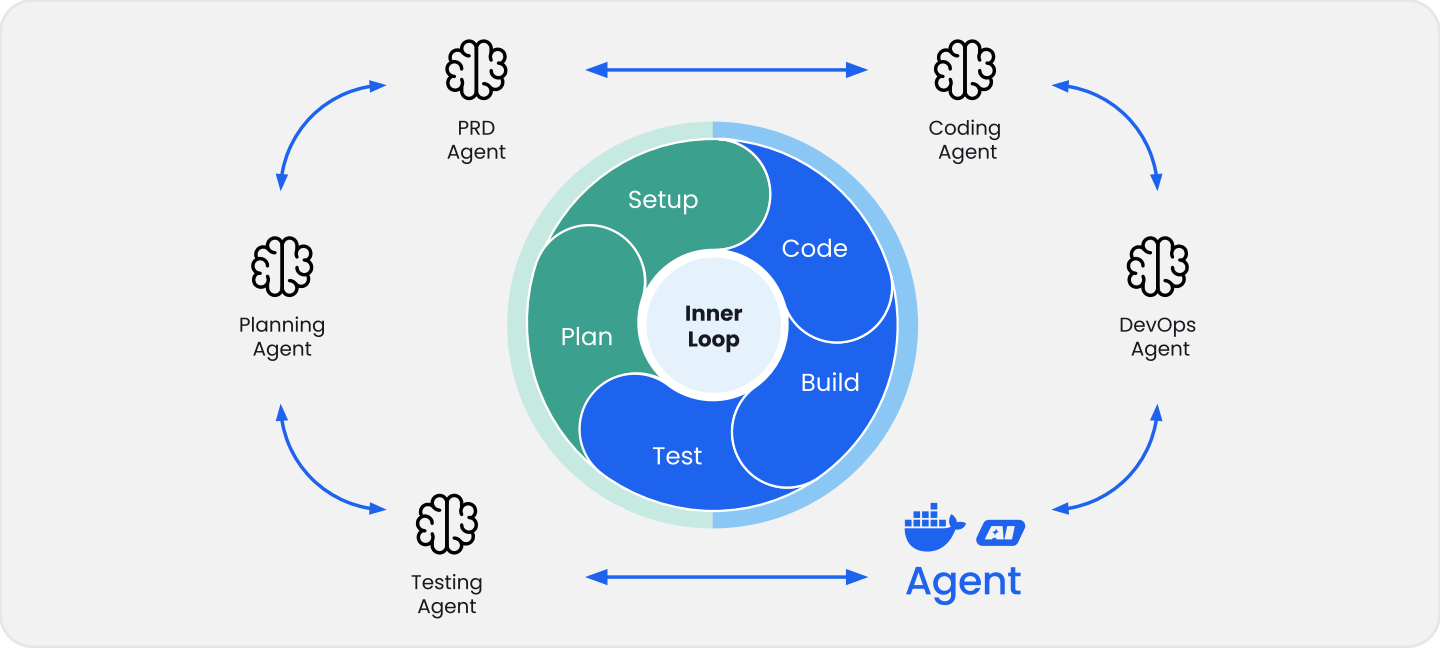



**Figure 9:** An example of an AI agent orchestrating other specialized agents in product development

## Chapter 3

# Innovation starts and thrives with the right partner

**Choose a partner who offers:**

**Trust**
Security, support, alignment

**Speed**
Quick from prototype to production

**Ecosystem**
Access to best practices, developer tools, verified content

Building successful GenAI applications isn't just about having the latest models or tools — it's about navigating complexity, uncertainty, and scale. That's where the right partner becomes critical. While it's tempting to go the DIY or open-source route, organizations often underestimate the depth of expertise, infrastructure, and coordination required to bring GenAI solutions to life while maintaining safety, security, and scalability.

A capable partner can accelerate your time to value, reduce operational risk, and help you avoid costly mistakes. They bring specialized knowledge, tested tools, and proven workflows that allow your teams to focus on what matters most: solving real problems with AI. Whether you're developing internal agents, customer-facing assistants, or complex data workflows, having a strong partner increases your chances of long-term success. Especially if the partner has a proven track record of guiding industries through disruptive change.

## What to look for in a partner

### 01 Trust

**Your partner should meet you where you are, providing security, performance, and ease of use while aligning with your specific requirements.**

**Why this matters:** Ongoing change is inevitable. As the AI ecosystem evolves rapidly with new models, APIs, and standards, your partner needs to adapt with you. Trust ensures that when something breaks, shifts, or needs to scale, you're working with someone who's aligned with your goals and invested in your long-term success, not just the initial implementation.

### 02 Speed

**Innovation moves fast, and so should your partner. They should deliver solutions and support quickly, adapting to change at the pace your business requires.**

**Why this matters:** An MIT study found that top business performers are both faster to market and innovative. They performed respectively 9.8 and 11.6 percentage points higher on net profit margin and revenue growth compared to industry averages.

### 03 Ecosystem

**The best partners are deeply connected to developer communities and industry leaders, giving you access to best practices and cutting-edge technologies.**

**Why this matters:** Ecosystem depth reduces integration time, accelerates onboarding, and ensures compatibility with future AI trends.

## Checklist: Questions to ask when evaluating a partner

**Use this list to guide internal discussions or vendor evaluations:**

- Does this partner have a proven track record of guiding industries through disruptive change?
- Can this partner meet your security, compliance, and governance needs?
- How quickly can they help you go from prototype to production?
- Do they integrate easily with the tools and platforms you already use?
- What kind of developer tooling and support do they offer?
- Do they have real-world case studies or references in your industry?
- Do they offer transparent pricing, SLAs, and documentation?

## Real-world example: Ingka Group x Docker

Ingka Group, the operator of IKEA Retail, is a giant in Home Furnishings and Retail with more than 482 stores across 31 countries.

### Challenges

Security played a critical role in Ingka's ML and AI development journey. With a global footprint and access to sensitive customer data, ensuring privacy throughout model training was non-negotiable. Their deployed models needed strong protections against tampering, and all dependencies had to meet high standards for vulnerability management.

### Solution

Docker's containerization and security features helped to protect Ingka Group's data and AI/ML models throughout the development and deployment processes.

- **Docker's containerization technology** ensured isolation and easier model management across different environments
- **Docker Scout** was used to analyze image layers and provided remediation suggestions, often recommending a more secure or up-to-date base image from **Docker Trusted Content**.

**Results**

Docker helped Ingka bring consistency and security to every stage of their ML pipeline — from model training to production deployment. By sourcing trusted, secure images from Docker's ecosystem, they moved faster, built smarter, and kept compliance on track.

With the right partner, building GenAI apps becomes easier, more secure, and highly efficient. This way, you can focus on innovation while staying ahead of the competition.

# Conclusion

AI is transforming the software industry, and companies that don't adapt risk falling behind. But with the right technology and strategic partners, businesses can seamlessly integrate AI into their workflows, boost productivity, and build innovative GenAI and agentic services that drive growth.

Using containers as the underlying technology shifts AI adoption from disruption to continuous improvement, making it easier to build, manage, and scale GenAI applications over time. The right partner delivers trusted technology, ongoing support, and innovation, and opens the door to a broader ecosystem and continuous learning. This speed advantage helps businesses stay ahead of the curve in an ever-evolving market.

**With containers as the foundation and a trusted partner like Docker, businesses can execute their AI strategy with speed and confidence. Docker's AI-ready platform enables enterprises to:**

- Empower developers to build GenAI apps and agents easily.
- Boost developer productivity with AI-enhanced workflows.
- Ensure security and compliance at every stage.

Every company is becoming an AI company. Containers are the key. Are you ready?

Visit our AI/ML blog to learn more about how you can get started with GenAI development with Docker today!

# References


1. **The ROI of GenAI: A global survey of enterprise adoption and value**
   https://cloud.google.com/resources/roi-of-generative-ai

2. **A visualisation of major large-language models (LLMs)**
   https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

3. **Enterprises lean on container solutions to deploy generative AI**
   https://www.ciodive.com/news/generaltive-ai-hybrid-cloud-containers-nutanix/739233/

4. **Survey: Generative AI Makes Tasks Simple, But Developing That AI is Anything But**
   https://newsroom.ibm.com/blog-survey-generative-ai-makes-tasks-simple,-but-developing-that-ai-is-anything-but

5. **DORA Report 2024 – A Look at Throughput and Stability**
   https://redmonk.com/rstephens/2024/11/26/dora2024/

6. **Going Faster Is Not Enough; Add Innovation to Outperform**
   https://cisr.mit.edu/publication/2023_0401_SpeedandInnovation_WeillBrechtWoerner