Coder

The Cloud Development Environment

ADOPTION REPORT

Message from the CEO

ROB WHITELEY

One thing is consistent in all of my customer conversations: Developers are the backbone of the digital era.

The code they write fuels new services, attracts new customers, and enables competition in new markets. But there's a catch. Developer productivity is staggeringly low—and for good reason. We've "shifted left" so many tasks that less than a third of developer's time is spent actually coding.

What if this could be fixed and 10, 15, or even 20% of a developer's time could be given back to them? Many companies with cloud development environments (CDEs) can accomplish this and realize enormous time and cost savings.

CDEs shift the developer's workspace from local machines to the cloud and reimagine most of the developer experience, from the earliest stages of writing code to deploying finalized applications. CDEs offer seamless collaboration, instant scalability, and reduced overhead by integrating numerous development tools—like the latest generative AI solutions—into a single cloud-based ecosystem.

CDEs, including Coder, are built for today's hybrid and remote environments. Rather than being confined to local workstations, CDEs make use of cloud computing (on-prem, in private clouds, or in public clouds) to automate workspace provisioning and shorten build times. Working in the cloud brings other advantages too, including more robust access controls, the ability to centralize source code and get it off laptops, and access to exotic compute requirements like GPUs.

In order to further explore CDEs' importance to the software development market, Coder conducted a survey of **223 developers** and business leaders globally from November 10 to 23, 2023. We focused on enterprises employing at least **2,000 workers** and **250 developers** and summarized their responses into the key data points and themes of this report. Among the leading findings, the survey revealed that **95% of participants are familiar with CDEs**, showing widespread awareness. Additionally, **66% of large organizations are currently using CDEs, highlighting their rapid growth despite being an emerging technology.** Why are CDEs experiencing such growth? Happier, more productive developers. Developers reported a 15% improvement in their quality of life from using CDEs, along with boosts in developer productivity, overall developer experience, and meaningful cost-savings for organizations.

With CDEs, as with any emerging technology, today's CIOs and CTOs need to avoid the pitfalls of "boiling the ocean." CDEs are implemented to solve a specific problem, often around providing access for third-party contract developers, giving data science teams access to AI and ML resources, or shortening the time it takes to onboard a new developer or switch an existing developer to a new project.

With so many use cases, it's critical for technology leaders to focus. This report highlights many CDE benefits, but we recommend building the business case for CDEs by focusing on four:

- Onboarding developers faster
- Securing source code
- Improving developer experience
- Cutting cloud costs

CDEs are a major milestone in the trajectory of software development. Beyond just offering a platform for coding, these environments epitomize the convergence of various technologies—new computing resources, workspaces as code, and generative AI to name a few—that bring the whole developer experience into the cloud.

I invite you to explore this report and learn how CDEs transform software development, offering insights into their increasing necessity for achieving efficiency, security, and cost savings.

INTRODUCTION

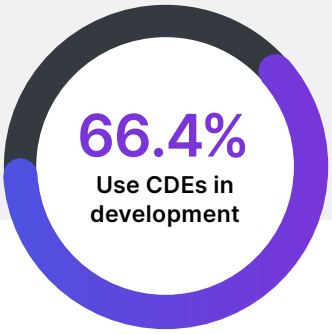
Evolution of Development Environments

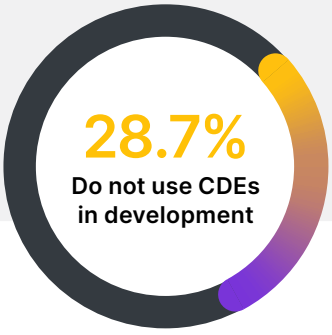
Cloud Development Environments (CDEs) are becoming the next frontier of programming productivity. Gartner® established CDEs as a separate category in the 2023 Hype Cycle™ for Emerging Technologies, including them among a list of promising technologies that are both delivering value now and will reach mainstream adoption within five to ten years.

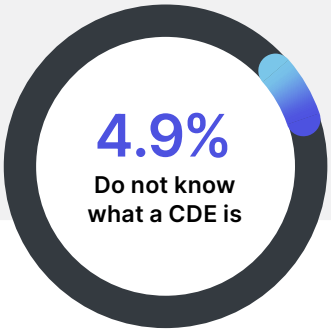
According to 2023 Gartner Hype Cycle report, “CDEs provide remote, ready-to-use access to a cloud-hosted development environment with minimal effort for setup and configuration. This decoupling of the development workspace from the physical workstation enables a low-friction, consistent developer experience.”

Gartner predicts that “by 2026, 60% of cloud workloads will be built and deployed using CDEs.” Our survey data supports this claim, showing that large enterprises are rapidly moving to the cloud with 66% of respondents indicating they already develop in CDEs.

CDE FAMILIARITY IN LARGE ORGANIZATIONS







Challenges That CDEs Solve

The emergence of CDEs represents a major milestone in the trajectory of software development, bringing with them an era of centralized, efficient, and cloud-centric workflows. They provide many benefits, including time and cost savings, increased developer productivity and experience, enhanced security and collaboration, expanded access to developer tools, scalability, and streamlined workflows. Beyond offering a platform for coding, these environments epitomize the convergence of various technologies that truly bring the whole developer experience into the cloud and the next echelon of software development.

In our survey, we asked respondents to indicate which benefits of CDEs they find most useful:

15.4%
Optimising the environment setup process

PERCEIVED BENEFITS

13.9%
Making environment management easier (onboarding and troubleshooting)

13.9%
Improve accessibility with remote access

12.6%
Improve security

12.4%
More flexibility in tool choices

11.7%
Ensuring compatability with production and staging environments

10.6%
Improve compatibility and reduce drift

8.7%
Access to server-grade computer resources

0.7%
I'm not sure/ I don't know

This is not an exhaustive list of all benefits of CDEs; however, this does highlight the key challenges developers and business leaders are addressing on a regular basis with this technology. We'll further explore the four benefits that proved to be the most popular in the survey: optimization of environment setup and onboarding, improvement to the developer experience through environment management and remote accessibility, security of companies' proprietary code, and cost savings for organizations.

Companies moving to CDEs can expect to generate tangible benefits across their operations. They should set goals, but they shouldn't try to accomplish every goal all at once. For the industry to evolve, they should instead focus on a few priority goals as they roll out the technology to their teams to ensure they are seeing and can justify ROI.

CDEs Simplify and Optimize the Onboarding Process

The move to remote work, accelerated by the pandemic, has complicated the onboarding process for employees across organizations. This is particularly true for developers, who shift to new projects upwards of five times a year and need access to a wide array of tools to do their jobs. Our survey shows that the average number of local tools a developer uses is greater than five, though some respondents indicate that they use up to 12 unique tools. We define tools as any piece of software that promotes developer productivity on the machine they use to write code, from IDEs to local VM managers.

In conventional enterprise environments, it can take days or weeks to equip each developer before any lines of code are written.

Skydio

"The time from setting up a new workstation to the first commit to production plummeted from a week to as little as an hour."

– Elliot Graebert, Director of Engineering, Infrastructure, Skydio

CDEs remove all the complexity from the onboarding experience. All the development tools, services, and infrastructure developers need can be made accessible through a web browser, or even by connecting through an extension on your local IDE. Often with a configured CDE, all IT departments have to do is give developers a URL and have them sign on. This cuts the onboarding time down to hours or even minutes.

As noted, onboarding isn't just a one-time task to be carried out at the time of hiring. Large enterprises typically will shift developers to new projects four-six times per year, each incurring an "onboarding" tax as they set up the right tools, languages, dependencies, and repositories.

Developers don't have to set up or manage their local environments. Plus, developers across teams and geographic locations can work on the same project at the same time, making it easier to collaborate, swap code, and stay in sync with each other.

Following are a few of the gains a personal finance company realized by moving to a CDE. The average time for setting up a Python coding environment with a complete set of packages dropped from 30 minutes to less than a minute while adding additional flexibility of hardware choices. **The company also reported a 97% improvement in build time efficiency using Coder's CDE.**

CDEs Keep Companies' Code Secure

Solving security threats is perceived as a critical concern in CDEs, and the data supports this notion. We asked developers why they rejected CDE products in the past, and five out of 10 tools were most frequently rejected due to a lack of security features. Overall, security was the #1 consideration when selecting tools, with 43.9% of respondents indicating it as a key feature when selecting CDEs.

It's no surprise considering how fast software supply chain attacks are escalating. A recent report by Cybersecurity Ventures projected the global annual cost of these attacks will grow 15% a year, from $46 billion in 2023 to $138 billion in 2031.

70% Security

10% Environment and IDE Support

28.7% Ease & speed of development

In a conventional development environment, contractors come in for three months and check out all source code on their laptops—but what happens when they leave? With a CDE, IT can eliminate the ability for anybody to download source code. No one can copy the files out into a separate environment, and there's no way for them to download the content. The only thing they see is the IDE that's running on a cloud instance. They can see and work on the code, but they can't pull that code out and put it on a device IT doesn't control.

If a contractor brings in their own device, that's also a security risk because they can download sensitive code and walk away. With CDEs, however, developers can bring their own device without issue. It doesn't matter what OS or what architecture the enterprise has as long as users have an approved browser. Organizations don't have to worry about source code walking out the door or someone stealing an unencrypted laptop. With a CDE, developers only have access to the IDE, working on storage and systems that IT controls.

While CDEs provide important source code protection, they provide additional security benefits as well. They meet the needs of security-conscious organizations requiring comprehensive auditing, role-based access control, user provisioning and deprovisioning, and data exfiltration to run zero-trust environments.

Palantir

said the ability to secure more CPU/RAM with the push of a button sped up the entire

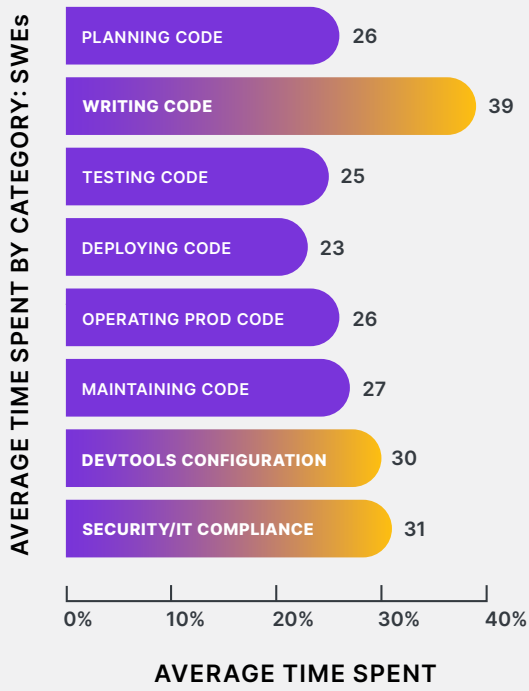
DEVELOP → BUILD → TEST

process across the board, cutting front-end build times by 78%.

CDEs Improve the Developer Experience

If there's one thing developers hate, it's friction in the development process. They want to spend their time coding, not getting hung up by ancillary tasks like onboarding or ongoing maintenance of code.

Based on our survey, developers find themselves burdened by other tasks, which are taking away from their time to code, including:



Other surveys find the same issues. According to a New Stack study, respondents spend 35% of their time managing code, including code maintenance (19%), testing (12%), and responding to security issues (4%).

According to a GitHub survey, developers spend the most time:

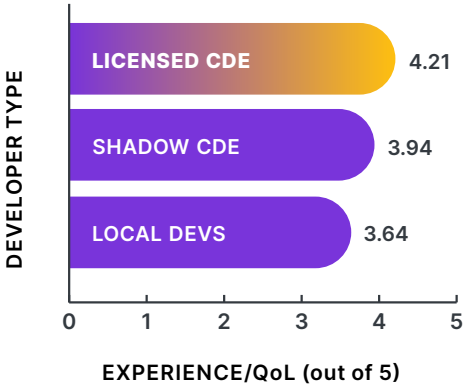
- Writing code (32%)
- Finding and fixing security vulnerabilities (31%)
- Getting feedback from end users (30%)
- Professional development and learning skills (30%)
- Regular asynchronous and team communication (30%)
- Specified blocks with no team communication (30%)
- Regular team meetings throughout the day (29%)
- Designing solutions to novel problems (29%)

Software company Palantir Technologies reports that CDEs help to speed up development cycles, reducing friction and leading to a better overall user experience.

"Since the workspaces live right next to our code repository and artifact servers, download times and request latency are very low," said a member of the Palantir development team. "For an example repository, this improved git clone times by 71%. While we expected faster development cycles in general, this was an unintended benefit."

Reports show developers put a high premium on positive experiences in their job environments. In one study by ZenHub, when asked to share how satisfied they are with their jobs, the majority of developers reported being mostly satisfied, with just 31% saying they were extremely satisfied.

Our survey found evidence of positive, consistent developer experiences, and the bars below show average developer experience by category. Each group only includes active users of the development solution. **Here, we see developers using licensed CDEs reported a 15% higher score in overall quality of work life than developers in conventional environments.**



They Cut Costs

CDEs do more than just make developers happy and strengthen the security of enterprises' valuable code. Shifting coding online saves companies' significant amounts of money as well.

To quantify the savings, we conducted a thought experiment using data from our survey, factoring in expected time savings enterprises can achieve in onboarding costs and general ongoing operations.

Let's start with onboarding. Assuming average developer densities of 35.4%, average software developer salaries of $144,000, and savings of 40 hours on time to first commit for each developer (a number provided by customer testimonials), organizations with 2,000 employees can save nearly $2 million in onboarding costs. **Savings for organizations with 10,000 employees skyrocket even further—up to nearly $10 million.**

ONBOARDING STAFF SAVINGS			
Employee Count	**Estimated Developers**	**Hours Saved Oboarding Full Developer Staff**	**Dollars Saved Onboarding Full Developer Staff**
2,000	~708	28,530 hrs	**$1,962,542**
5,000	~1,771	70,840 hrs	**$4,911,630**
10,000	~3,543	141,720 hrs	**$9,826,033**

Once developers are onboarded, it's reasonable to assume that each developer would save at least two hours of time each week on improved build times and reduced tool maintenance. Using these numbers, the same smaller organization with 2,000 employees saves over $5 million annually, and its larger counterpart nets out to more than $25.5 million in annual savings.

Those savings are welcome additions for organizations to either improve their bottom lines or to free up funding to bring on additional developers.

SALARY FOR ADDITIONAL DEVELOPERS			
Employee Count	**Estimated Developers**	**Developer Hours Saved Yearly**	**Developer Salary Saved Yearly**
2,000	~708	73,632 hrs	**$5,105,211**
5,000	~1,771	184,184 hrs	**$12,770,238**
10,000	~3,543	368,472 hrs	**$25,547,687**

Onboarding aside, CDEs save time and money in other ways too. CDEs remove the burden of developers having to switch from platform to platform, sometimes multiple times a day. Also, by shifting burst capabilities to the cloud, enterprises can avoid the cost of having VMs run 24/7.

Elliott Graebert, director of engineering, infrastructure, at drone manufacturer Skydio, estimates that his company reduced its cloud computing costs for development environments by an astounding 90% simply by automating the shutdown of unused VMs.

CDEs also can lower the costs of requiring people to manage in-house solutions or automated tools for dev environments. Before joining Skydio, Graebert was Head of Internal Infrastructure at finance company Palantir where his team developed an in-house web portal for virtualized desktop environments. The project took 3 developers to build, and it was in use for years. In his experience, "while it started off as an exciting software project, it turned into a maintenance burden that needed constant attention."

The CDE Landscape

Now that we've outlined the primary benefits of CDEs, gaining an understanding of the CDE landscape will help identify which tools are best suited for your organization.

When organizations migrate to remote development without the help of CDEs, platform engineering teams and DevOps groups often set up virtual machines in the cloud to run 24/7 and manually configure developer access, a solution we call "Shadow CDEs." While the upfront cost of self-maintained VMs is cheaper, they have no native support for tooling, nor methods to minimize cloud costs without the assistance of a licensed product.

This wide roster of CDEs includes a number of variations, each improving unique aspects of the migration to remote development, which makes determining which platform is the best fit for your organization challenging.

A primary consideration when comparing CDEs is the trade off between level of configuration and ease-of-use. Some development teams manage a diverse set of bespoke tools and infrastructure requirements and require flexible CDEs that can smoothly integrate with their internal platforms. Other teams need CDEs that are focused on improving the accessibility of computing while supporting simpler workflows that require little to no setup from the end user.

There are differences in the amount of tooling support certain types of CDEs can provide. Even further, there's a fundamental difference in hosting status: whether a CDE is self-hosted or available through a SaaS subscription.

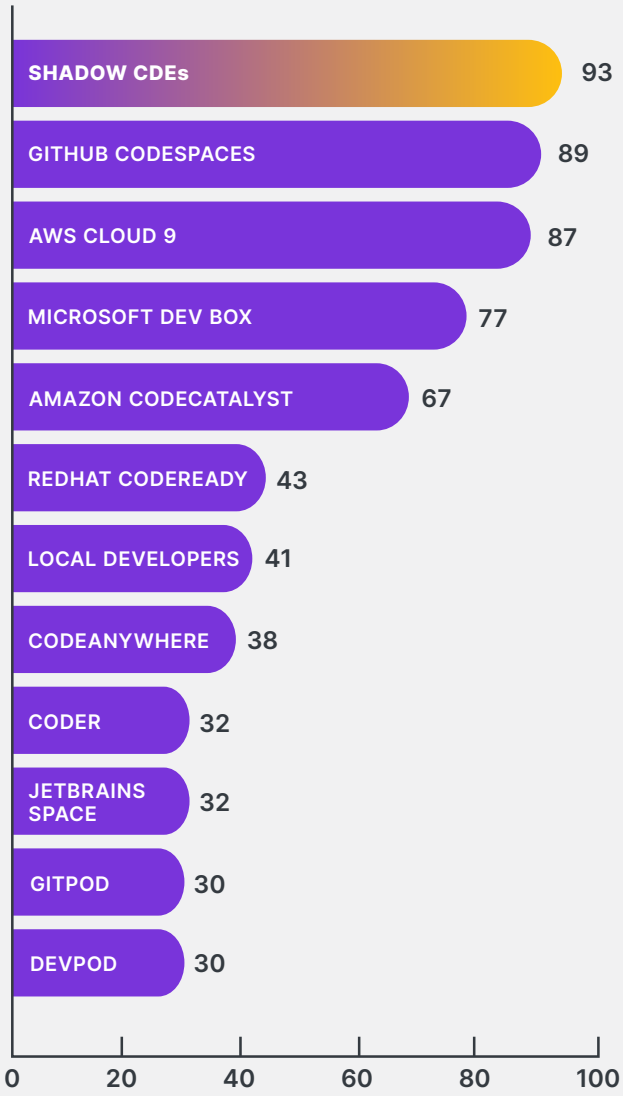
SELF-HOSTED CDEs

Self-hosted CDEs, which require organizations to both host and run their own environments, are preferred by more sophisticated early adopters who need the ability to exert tighter controls over their environments. Security tends to be the top reason for enterprises to self-host their CDEs, since it keeps the IP on the enterprise's infrastructure. That allows IT departments to put access controls in place and have stronger oversight over what developers are working on.

SaaS CDEs

SaaS CDEs are designed for development teams that need quick, reproducible development environments in the cloud with minimal infrastructure configuration. SaaS development environments are prepackaged CDEs written by an external company, and they don't require a platform engineering team to set up the platform internally.

Interestingly, our survey results show that these "Shadow CDEs" are still the most prevalent implementation of remote development tools:



Comprehensive CDEs

Companies that want a full-featured, out-of-the-box coding environment—and are willing to pay for it—are finding good fits with comprehensive cloud development platforms like *Amazon Code Catalyst* and *JetBrains Space*.

These platforms manage complete end-to-end software development in the cloud, from setting up development environments to deploying applications. These tools expect developers to live entirely within their proprietary ecosystems for all aspects of development. This makes them more beneficial for organizations that already depend on a single vendor for each step in the development cycle.

The main benefit of this type of environment is that it comes loaded with all the essential tools for everyday tasks such as version control, project management, database tools and unit testing. They're also highly customizable within the limits of their ecosystem, giving developers the ability to personalize their coding environments to meet their preferences.

The drawbacks? They can be expensive. Plus, they typically require a steep learning curve, and they may require high-performance machines to run effectively.

Collaboration-Focused Platforms

Collaboration-focused platforms such as *GitHub Codespaces* and *Microsoft Dev Box* work well for companies that have developers working closely with DevOps teams. The platforms build off of and integrate with existing DevOps systems, such as *GitHub* and *Azure*, to provide development environments that live closely to existing services. They are focused on integration with a small ecosystem but don't solve the entire development lifecycle like the comprehensive platforms. They're often delivered as SaaS alongside their related tools.

One of the key benefits of these platforms is the ability to spin up task-based developer environments in the cloud from the existing project context, such as a *Git* repository or an *Azure* deployment. Once you're done, you just close them. This is a huge boost for productivity. Development environments can be configured as code, allowing them to be version controlled, reproducible and automatable. The downside is they're complex to configure and hard to use for developers without extensive experience and advanced skill sets.

Integrated Development Environments in the Cloud

IDEs like *AWS Cloud 9* and *Codeanywhere* allow platform engineering teams to provide developers with unique proprietary IDEs in the cloud. They provide independent tools that are prepackaged for development on any machine, focusing on low set-up costs and streamlined interfaces. They're considered great collaboration tools, with embedded features like chat, voice, and video calls right in the IDE. With an API, they can integrate code environments right into a user's application without the overhead of integrating your own infrastructure or hardware.

Flexible Enterprise Cloud Development Environments

Tools provided by *Coder*, *GitPod*, and others create CDEs for developers on companies' existing infrastructure, optimizing for diverse cloud providers, and tooling support. They typically offer a self-hosted solution for enterprises, allowing for maximum security with the help of an established DevOps/platform engineering function. They tend to be optimized for large organizations. These vendors do not offer independent IDEs, instead allowing companies to use whatever IDEs and tooling they are already comfortable with.

INTEGRATED DEVELOPMENT ENVIRONMENTS (IDES) IN THE CLOUD

- Can provide independent tools that are prepackaged for development on any machine
- Offers low set-up costs and streamlined interfaces
- Weak in terms of compatibility, but strong in ease of use
- Examples: *AWS Cloud 9, Codeanywhere*

DEVOPS AND COLLABORATION-FOCUSED PLATFORMS

- Integrate with existing DevOps systems
- Focused on integration with a small ecosystem
- Unable to solve the entire development lifecycle like comprehensive platforms
- Examples: *Microsoft Dev Box, GitHub Codespaces, DevPod*

COMPREHENSIVE CLOUD DEVELOPMENT PLATFORMS

- End-to-end software development in the cloud
- Only requires configuration for projects
- Developers live entirely within their proprietary ecosystems
- Examples: *Amazon CodeCatalyst, JetBrains Space*

FLEXIBLE ENTERPRISE CLOUD DEVELOPMENT ENVIRONMENTS

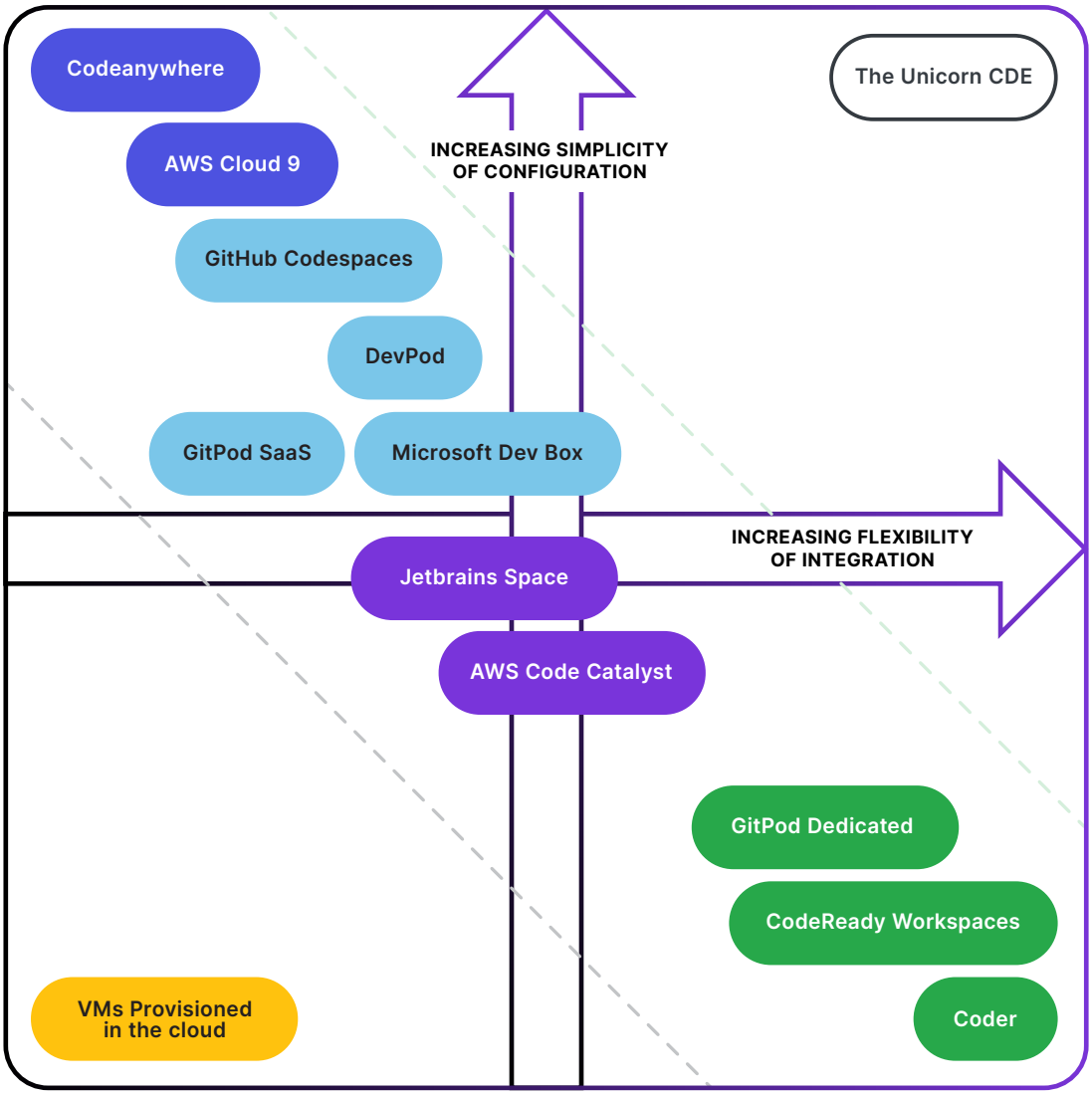
- CDEs for developers on companies' existing infrastructure
- Uniquely self-hosted, allowing for maximum security at a lower cost
- No independent IDEs; users have flexibility to use their tooling of choice
- Examples: *GitPod Dedicated, Coder, DevPod, CodeReady Workspaces*

Simple SaaS CDEs

Out-of-the-box simple solutions for SaaS. Lack security and infra benefits while gaining simplicity.

The Unicorn CDE

The ideal CDE integrates with everyone's infrastructure and requires no setup cost.



Shadow CDEs

Costly for setup, maintenance, and require major investment to enable integration.

Large Enterprise CDEs

These solutions provide maximum flexibility, but requires teams to manage infrastructure and developer tools. Suitable for large organizations.

KEY

- Unicorn CDE
- IDEs in the cloud
- DevOps, Collaboration Focused CDEs
- Comprehensive CDEs
- Flexible self-hosted
- Shadow CDEs

Boundary Line Explanation

If CDEs fall below the gray line, they are not an improvement to Shadow CDEs. But no tool can rise above the green line.

“CDE is a packaging exercise,” Redmonk analyst James Governor wrote in his report declaring 2023 as “The Year of the Cloud Development Environment.”

“As I often say,” Governor added, “the best packager in any tech wave wins and wins big. And I think what we’re currently looking at is packaging up how developers do their work and just trying to make that really, really easy.”

As we’ve seen in this report, CDEs provide a set of tangible benefits, all wrapped up in one easily configurable package:

- CDEs speed up and simplify the onboarding process for developers from several weeks down to a matter of minutes.
- CDEs keep companies’ code secure, removing the specter of developers downloading code on a USB and either sharing it or losing it.
- CDEs improve the overall developer experience, removing time and friction from everyday tasks.
- CDEs save companies time—an estimated 141,000 onboarding hours and nearly 370,000 hours in productivity gains for a 10,000-employee enterprise with 3,500 developers.
- CDEs save companies money—more than $35 million a year for the same-sized enterprise.

This is what CDEs do today: create self-serve environments that developers can access from anywhere without having to host code on their laptops or get DevOps or platform engineering involved in their processes. By expressing development workspaces as code, CDEs pull the last part of development —the developer workstation—into the cloud era.

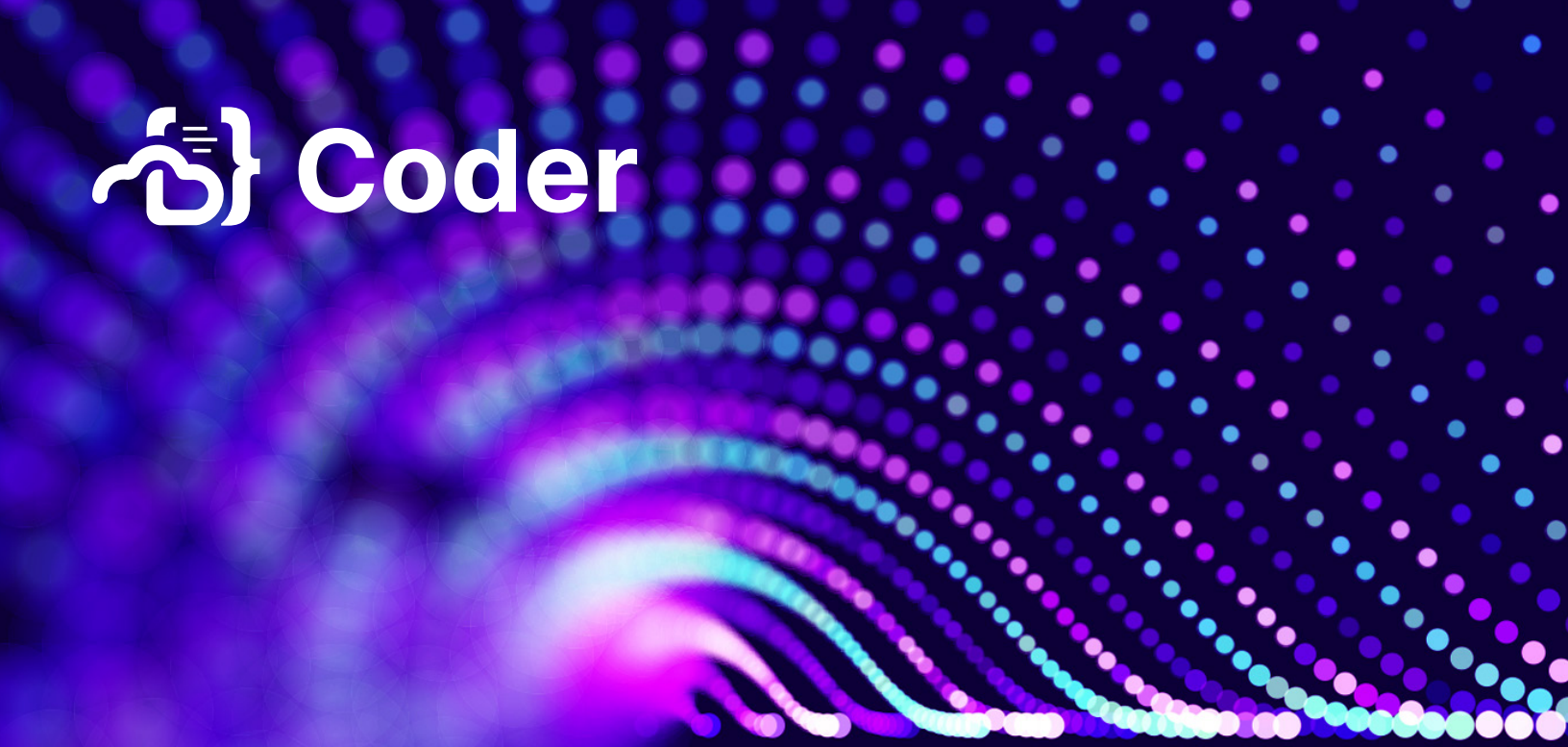
What will CDEs do in the future? An obvious future benefit will be to pull the developer workstation, and the enterprise itself, into the age of AI.

CDEs’ capabilities match up well with companies’ moves into AI. They’re an essential component of machine learning workloads, since local hardware will never have the GPU capacity to support training for large AI models. They keep developer workstations geographically close to data centers, shortening the time to migrate data sets to the machines that run these AI jobs.

Plus, CDEs serve the needs of the players in the pipeline of machine learning-powered data analysis, each with unique demands for tools and computation. Software engineering teams need IDEs, DevEx tools, fast build times, and smooth CI/CD integrations to build the product from which data is collected. Data engineers need IDEs and close proximity to the deployed product for quick iteration and scraping to set up collection systems to scrape activity from those products. Finally, data scientists need analysis-focused tools and access to high-grade GPU workloads to train machine learning models and optimize the product through analysis.

Without CDEs, these roles have to synchronize all their tools and pipelines independently, leading to costly configuration and devtool maintenance for the end developer. If these tasks are instead performed on CDEs, a centralized team can configure each team’s tools and provide them with a unique level of compute. This leads to more time for the developers and data scientists to focus on their code.

CDEs provide an environment where enterprises can maximize productivity and keep control—now and in the future, in the age of AI.



To learn more about Cloud Development Environments, check out the Coder **blog**. To see Coder in action, learn how customer **Skydio** succeeds with CDE technology.

> **If you're ready to take the next step in your CDE journey, contact us today.**