Coder

An Enterprise Buyer's Guide to

Cloud Development Environments

Comparison of Coder, GitHub, Amazon, Microsoft, Red Hat, and Gitpod cloud development environments

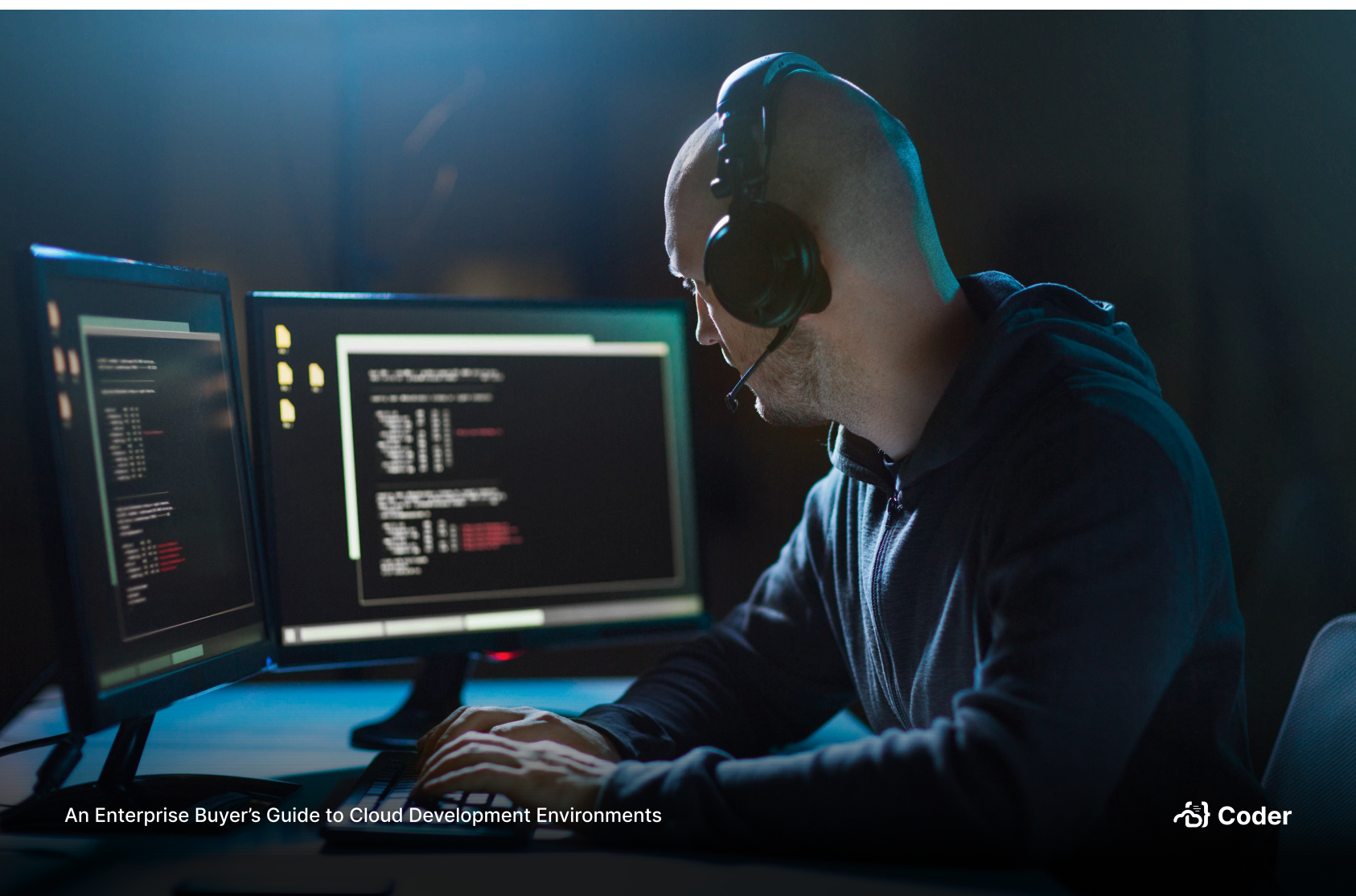
Table of contents

The Last Frontier of Cloud Migration: Development Environments

Enterprises have embraced cloud migration across nearly every aspect of their software delivery pipeline. From source control to CI/CD, security tooling to production infrastructure – almost everything has found its place in the cloud.

Yet, there's one critical area that has lagged behind: **development environments**.

In many organizations, development environments still exist on local machines, virtual desktops, or isolated virtual machines. Why? Developers and data scientists hold their environments close to their hearts. Development environments have traditionally been highly customized, personal spaces that each developer tailors to their specific workflows. Developers have lacked compelling alternatives to legacy development environments for years despite the broader industry's shift to the cloud.



Not a New Idea, but a New Era

Cloud Development Environments (CDEs) have been around for the better part of a decade, introduced as a solution to centralize, secure, and accelerate developer workflows. However, their early iterations failed to gain widespread adoption. Developers weren't willing to trade the familiarity of their local environments for the complexity or perceived performance hits of early CDEs.

But the landscape has changed.

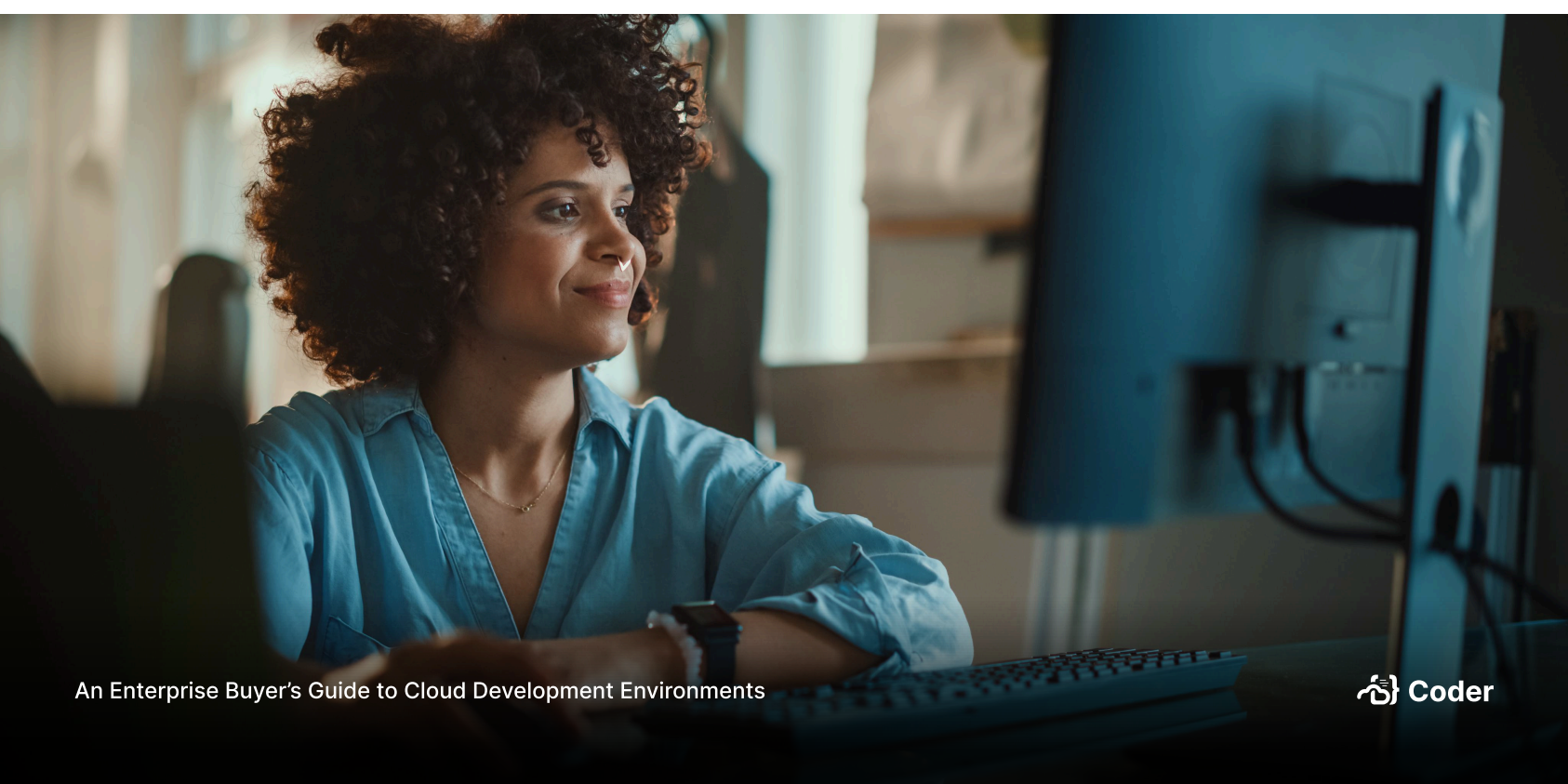
Recent advancements in networking, containerization, and cloud infrastructure have enabled us to deliver CDEs that not only match but surpass the capabilities of traditional development environments. This evolution has turned CDEs from a niche concept into a viable – and now preferred – replacement for local environments and virtual desktop infrastructure.

What CDEs Solve

Most CDEs deliver three core promises:

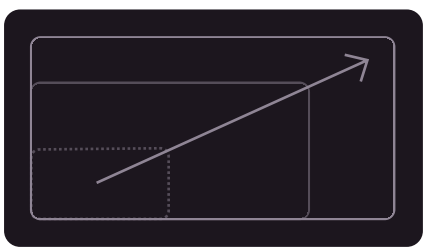
- **Security and Governance** – CDEs keep source code and data centralized in the company's cloud, enhancing security and reducing the risk of exposure on decentralized developer machines. IT teams can instantly apply patches, updates, and policies across all environments without relying on developers to update their local setups.
- **Developer Productivity** – CDEs simplify onboarding and ensure consistency across teams. Developers can start coding faster without spending days or even weeks setting up their local environments, allowing them to focus on solving meaningful problems rather than managing tools and configurations.
- **Cost Control** – Cloud resources are managed more efficiently, reducing waste, particularly with ephemeral environments that spin up and down as needed. CDEs also help rationalize tooling by consolidating the platform, driving further cost savings.

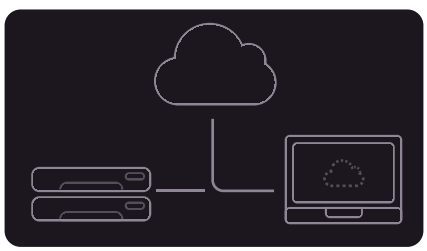
While these benefits are largely universal across most CDE solutions, **how they deliver these benefits varies**. The true inflection point in the market is happening now. Technology has evolved to a place where CDEs are no longer a compromise but a true improvement over local development environments and VDI.



The CDE Landscape

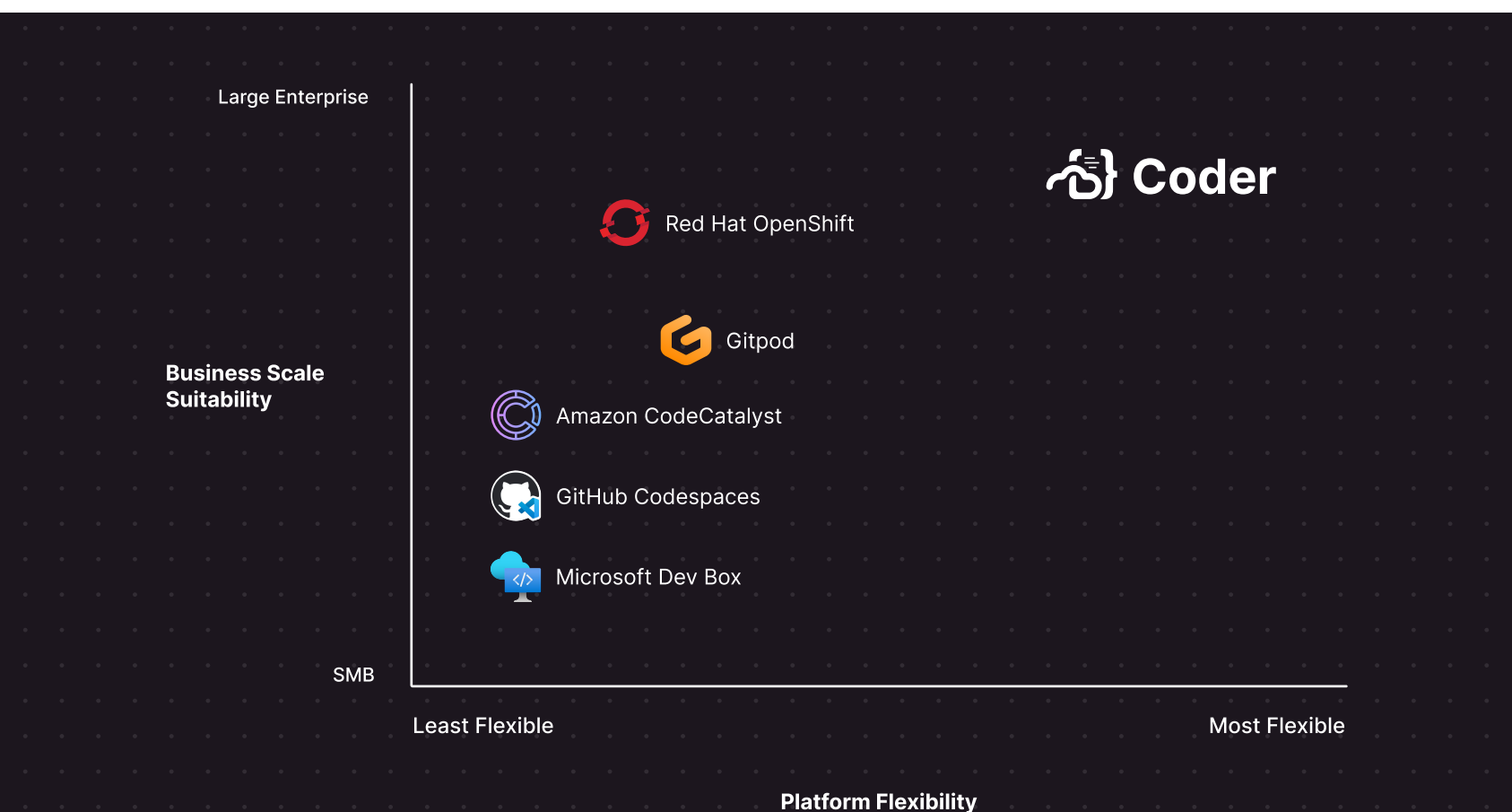
The CDE landscape can be mapped by evaluating two critical factors: business scale suitability and platform flexibility.





Business scale suitability focuses on how well a CDE supports different organizational sizes – from small teams to large enterprises. This includes assessing security capabilities, feature maturity, and the ability to handle the demands of enterprise-scale development teams with high availability and reliability.

Platform flexibility examines the degree to which a CDE can adapt to different infrastructure needs. This includes the ability to self-host, self-manage, and operate across multiple cloud environments and whether the platform is open source and capable of being deployed on-premises.



CDE Evaluation Criteria

When assessing CDEs, enterprises consider several factors that influence both operational efficiency and long-term performance and scalability. The following criteria are important in determining which platform can best meet business requirements, including flexibility, security, developer experience, cost-effectiveness, and integration with existing infrastructure.

- **Open Source** – Does the CDE offer a fully open-source platform, enabling full transparency, community-driven improvements, and the flexibility to modify the platform to meet specific needs?

- **Delivery Model** – How is the platform delivered? Some CDEs are offered as SaaS solutions, managed entirely by the provider, while others are available as self-hosted options, allowing businesses to deploy and manage the platform on their own infrastructure.

- **Maintenance Model** – Is the platform easy to maintain in-house, or does it require external access for vendor support? This criterion considers whether organizations can handle updates, patches, and configurations on their own, ensuring autonomy.

- **Data Sovereignty** – Does the CDE provide the ability to control where data is stored? This is critical for organizations with stringent data residency and compliance requirements, particularly in regulated industries.

- **100% Offline/Air-gapped Support** – Does the platform offer complete isolation of development environments, up to and including physical separation from external networks? Full isolation ensures that environments are separated at the infrastructure level, which is important for security and avoiding cross-environment interference.

- **Cost Model** – What is the cost structure of the platform? Consider whether the pricing is based on a subscription model, pay-per-use, or tied to the consumption of cloud resources. The cost model should align with the organization's financial goals and scalability plans.

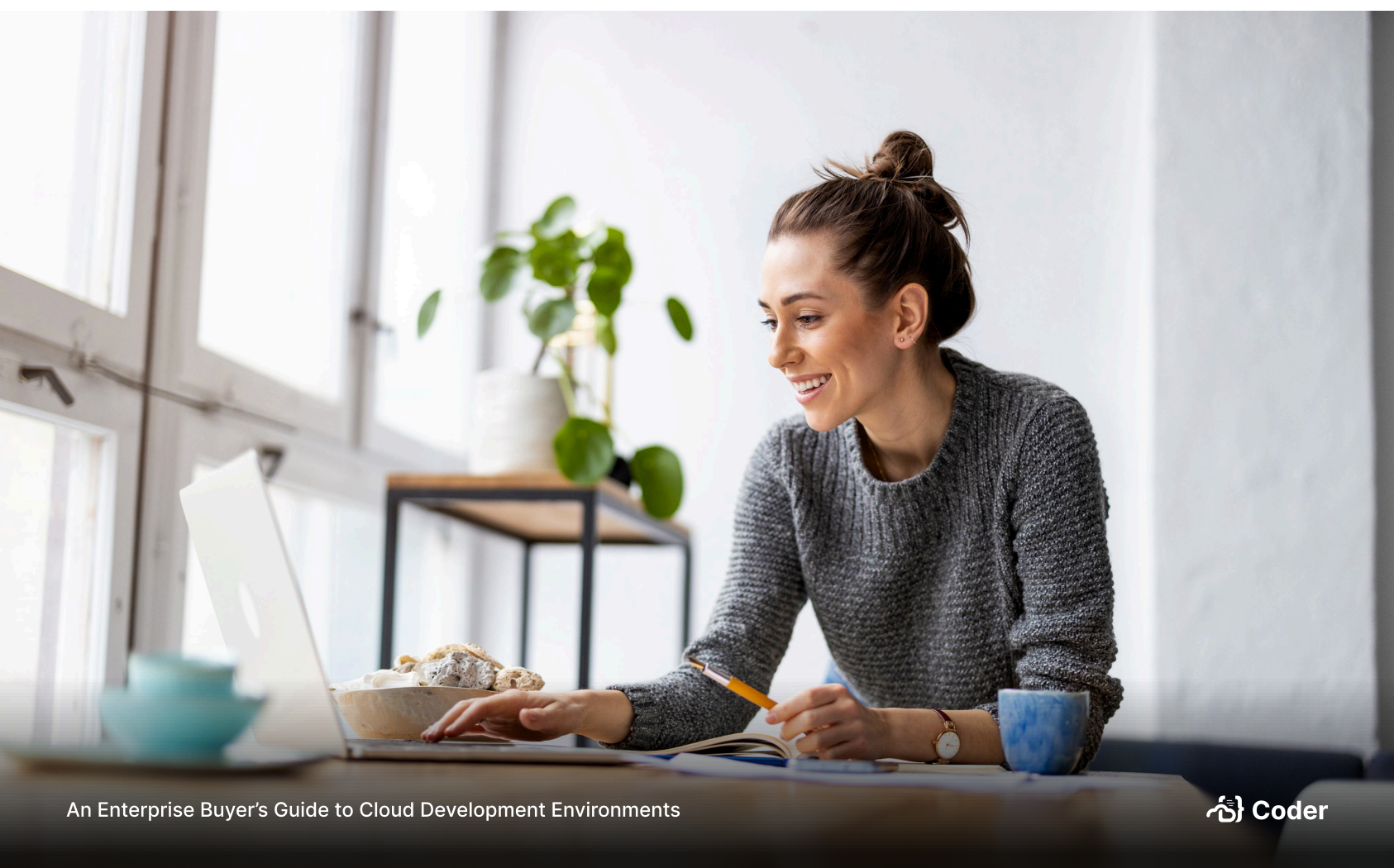
- **Multi-cloud Capable** – Can the CDE operate across multiple clouds (e.g., on-prem, AWS, Azure, Google Cloud), or is it locked into a single provider? Multi-cloud capability offers flexibility and reduces reliance on any one vendor.

- **Workspace Platform** – What underlying platform does the CDE support for workspaces? This could be virtual machines (VMs), Kubernetes pods, or Docker containers. The workspace platform directly impacts performance, scalability, and compatibility with existing infrastructure.

- **Workspace Extensibility** – How extensible are the development environments? This criterion examines whether the CDE allows customization and integration of additional tools or services to meet specific workflow needs.

- **Workspace OS** – What operating systems are supported for the development environments? Consider whether the CDE supports the required range of operating systems, such as Linux, Windows, or macOS, based on the organization's needs. Most CDEs only support Linux workspaces.

- **Workspace Tooling** – What built-in tooling or integrations does the platform offer? Evaluate how well the CDE supports developer tools and IDEs, and whether it supports the tooling for existing version control systems, CI/CD pipelines, and other key parts of the development workflow.



Comparing Evaluation Criteria

	Coder	GitHub Codespaces	Amazon CodeCatalyst	Microsoft Dev Box	Red Hat OpenShift Dev Spaces	Gitpod
Open Source	Fully open source, including Premium features	Closed-source integrated with GitHub's ecosystem	Closed-source integrated with AWS services	Closed-source integrated with Microsoft Azure	Open source via Eclipse Che	Fully open source
Delivery Model	Self-hosted for maximum control over infrastructure	SaaS only, hosted and managed by GitHub	SaaS only, hosted by AWS	SaaS only, hosted by Microsoft	Self-hosted on OpenShift	SaaS or self-hosted on AWS
Maintenance Model	Self-managed for autonomy over updates and configuration	Vendor-managed by GitHub	Vendor-managed by AWS	Vendor-managed by Microsoft	Self-managed	Vendor-managed, Gitpod handles application updates & security patches
Data Sovereignty	Complete control over where data is stored in your cloud(s) or on-premises	Limited control over data; all data is stored within GitHub's infrrastructure, primarily on Microsoft Azure	Limited control over data; all data is stored within AWS infrastructure, giving users some flexibility within AWS regions	Limited control over data; all data is stored within Microsoft's infrastructure, giving users some flexibility in terms of regions	Complete control over where data is stored in your cloud(s) or on-premises	Control over data within the limits of Gitpod's required AWS infrastructure
100% Offline Support	Supports fully offline operation for air-gapped environments	No offline support	No offline support	No offline support	Fully supports offline operation for air-gapped environments	Self-hosted requires access for vendor-managed updates.
Cost Model	Annually per user, per deployment	Usage-based tied to compute, storage, and time used	Monthly per user, plus usage	Usage-based tied to compute, storage, and time used	Based on Red Hat's subscription model, with additional costs for infrastructure resources	Usage-based for SaaS, monthly per user for self-hosted
Multi-cloud Capable	Deployable across any cloud, hybrid, or on-premises environment	Supports Microsoft Azure infrastructure only (via GitHub)	Supports AWS infrastructure only	Supports Microsoft Azure infrastructure only	Deployable across any cloud, hybrid, or on-premises environment	Supports AWS infrastructure only
Workspace Platform	Supports VMs, Kubernetes, and Docker for flexible workspace provisioning	Supports Docker only	Supports VMs only	Supports VMs via VDI	Supports OpenShift pods	Supports Kubernetes-based environments only
Workspace Extensibility	Supports Terraform templates and Dev Containers, allowing users to incorporate any Terraform use case and tailor their own workspaces	Supports Dev Containers only	Supports pre-configured devfile.yml blueprints on free tiers, while enterprise tiers enable custom blueprints	Supports management via Azure Web UI, primarily focused on integrating with Microsoft tools and services	Supports devfile.yml	Requires propriety gitpod.yml, similar to Docker-compose
Workspace OS	Linux, macOS, Windows	Linux	Amazon Linux, Windows Server 2022	Windows	Linux	Linux
Workspace Tooling	Works with any tool that the workspace OS supports (Linux, macOS, Windows)	Supports Linux tooling	Works with AWS products and tooling supported by workspace OS	Supports Windows tooling	Supports any tool that can be containerized in OpenShift	Supports Linux tooling

Comparing Ecosystem Support

	Air-Gapped	AWS	Google	Azure	Docker	Kubernetes	Terraform	Windows	Linux	macOS	VS Code	Visual Studio	IntelliJ	JupyterLab	Cursor AI	GitHub	GitLab	Bitbucket
	Infrastructure							Operating System			IDE					Git		
Coder	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
GitHub Codespaces				✓	✓				✓		✓		✓	✓		✓		
Amazon CodeCatalyst		✓			✓			✓	✓		✓		✓			✓	✓	✓
Microsoft Dev Box				✓	✓			✓			✓	✓	✓	✓	✓	✓		
OpenShift Dev Spaces	✓	✓	✓	✓	✓	✓			✓		✓		✓	✓		✓	✓	✓
Gitpod	✓	✓			✓	✓			✓		✓		✓	✓	✓	✓	✓	✓

Why Enterprises Choose Coder

Every organization has unique requirements for a CDE, and the expanding CDE landscape offers plenty of options. Smaller teams without strict security or governance needs may prefer a SaaS platform on managed infrastructure, while larger enterprises typically require stronger security, greater platform extensibility, and seamless integration with their existing systems.

Enterprises choose Coder as their CDE platform because it is **self-hosted**, self-managed, and multi-cloud. Hosting flexibility allows organizations to run Coder in any cloud, including fully air-gapped on-premises[1] environments. Because Coder is **self-managed**, platform teams control when and how updates are rolled out. This sets Coder apart from platforms like **Gitpod**, which is vendor-managed and is limited to running on AWS infrastructure. Coder's adoption in the enterprise market has largely been driven by its self-hosted, self-managed platform, which provides platform teams with full control and scalability.

Being fully open source[2] also appeals to companies that want the flexibility to customize their deployments and actively contribute to the growth of Coder's community and platform. In addition, Coder is both **IDE-agnostic** and **OS-agnostic**, supporting a wide range of development tools and environments. In contrast, cloud-native platforms from **GitHub**, **Amazon**, and **Microsoft** are closed-source and vendor-managed, offering limited flexibility. These platforms are typically restricted to the operating systems and developer tools that align with their respective ecosystems, limiting the options available to platform teams.

Organizations sometimes compare Coder to **Red Hat OpenShift Dev Spaces**, both self-hosted and self-managed platforms. While Dev Spaces can be a viable option for enterprises heavily invested in OpenShift, it requires full commitment to the Red Hat ecosystem. In contrast, Coder is unopinionated and integrates with every infrastructure and tool stack, including OpenShift, allowing developers to move beyond incumbent systems that hinder a superior developer experience.

Additionally, Coder is highly extensible and **integrates well with existing tools and infrastructure**, so organizations don't need to overhaul their stack to implement CDEs. It works with existing authentication, image registry, and monitoring systems, allowing enterprises to maintain their current ecosystem.

More than just a platform for running containers, Coder leverages Terraform infrastructure-as-code to provision additional software-defined infrastructure like networks, storage, and clusters across multiple platforms and clouds. This reflects Coder's focus on using proven, open-source technologies rather than reinventing the wheel.

For organizations that prioritize security, control, and scalability, Coder provides a reliable and flexible CDE platform. Here's why organizations across different sectors rely on Coder:

Global 2000 Companies

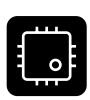
Large organizations need a CDE that can scale to tens of thousands of users while maintaining high performance and reliability. Coder is built for this level of scale, offering a distributed and resilient platform that can handle extensive workloads without sacrificing performance. It's designed to support large teams across different global regions.

Large Financial Institutions

For financial institutions, compliance, data sovereignty, and security are top priorities. Coder meets these strict requirements by providing a secure, self-hosted platform that keeps sensitive data fully under the organization's control. It integrates with existing systems while ensuring compliance with regulatory standards for data privacy and security.

Government and Intelligence Agencies

Government and intelligence agencies in the US and across Europe require 100% air-gapped environments to safeguard their systems from external threats. Coder operates in completely isolated environments, giving agencies the highest levels of control and security. As a fully open-source platform, Coder accelerates the security accreditation process, including compliance with region-specific standards, such as the Authority to Operate in the US and the UK Government's National Cyber Security Centre guidelines and Data Protection Act.



Tech Innovators

Tech innovators need flexibility to adapt to evolving demands. Coder's open-source foundation allows organizations to customize the platform to fit their needs, integrating with any infrastructure or toolset. This enables tech companies to build and modify their development environments without being limited by vendor constraints.

Supporting More Than Just Developers

Coder isn't limited to development environments for developers alone; it also opens up valuable use cases for **data engineers**, **data scientists**, and **analysts**. These teams can securely access cloud-based tools, like Jupyter Notebook, and large datasets without needing expensive, resource-constrained local hardware. By leveraging cloud computing, they efficiently tackle model training and data analysis while adhering to security and compliance requirements.

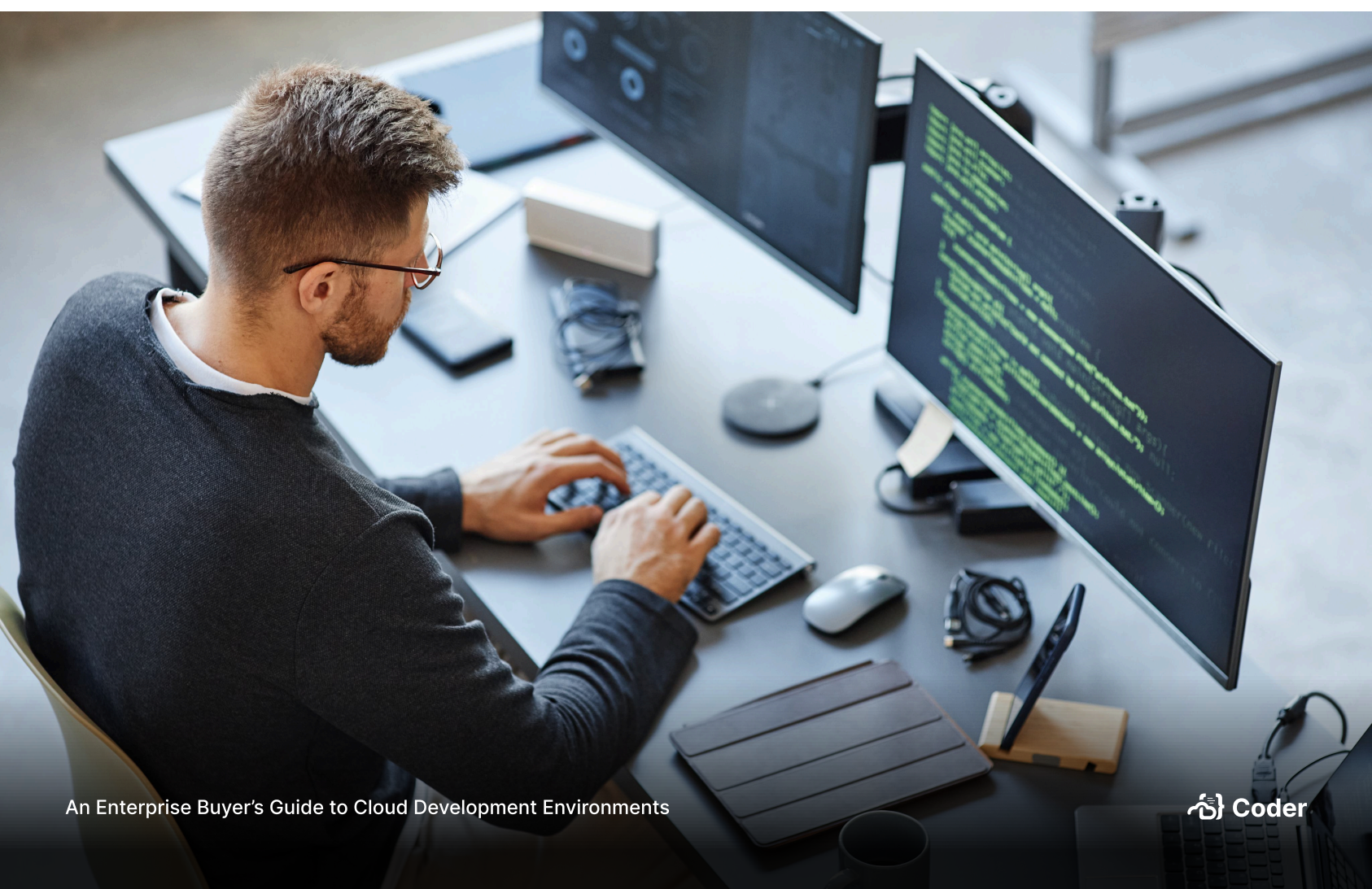
One example is a major investment bank that used Coder to shorten feedback loops between analysts and traders. By connecting both groups directly to the same models in real time, Coder removed the need for complex pipelines, allowing analysts and traders to quickly test, iterate, and respond to market changes.

A Modern Alternative to Legacy VDI

For organizations looking to move beyond traditional VDI solutions, there is a more cost-effective[3] and developer-friendly approach.

Using Coder alone elevates an organization's security posture by centralizing source code in secure, cloud-based environments that are fully under the organization's control. Adding Island's Enterprise Browser[4] provides a next-level layer of data loss prevention by enabling browser-level controls that prevent the copying and pasting of secure source code and data from browser-based IDEs to a developer's local desktop. Together, Coder and Island represent a shift in security approach, delivering the protection of VDI via a modern and developer-friendly model.

Unlike VDI, which often comes with escalating costs and a poor developer experience due to lag and performance issues, Coder and Island provide a solution that addresses these pain points. Developers can work efficiently in a cloud-native environment while enterprises gain the security and compliance needed without the heavy costs and frustrations associated with traditional VDI.



Finding the Right CDE for Your Organization

Choosing the right CDE depends on identifying what matters most to your organization, whether you're a hobbyist, a small business, or a large enterprise. Every company has unique needs when it comes to security, scalability, flexibility, and integration with existing systems. The growing landscape of CDE options offers plenty of choices, but not every platform will be the right fit for every company. Each CDE has its strengths, making it important to assess the market carefully to find the platform that aligns with your specific requirements.

Contact us[5] to explore the cost, security, and developer productivity benefits of Coder.

---

1 https://coder.com/blog/security-stability-scalability-with-coder
2 https://github.com/coder/coder
3 https://coder.com/success-stories/jbhunt
4 https://coder.com/blog/the-modern-alternative-to-vdi-for-developers
5 https://coder.com/contact/sales