COMMISIONED BY

Buyers Checklist: Assessing Container Development Environments

DEVOPS

Table of Contents

This GigaOm Buyers Checklist is commissioned by Docker.

1. Executive Summary

Today, software delivery is driving business outcomes. As organizations increasingly adopt cloud-native applications based on containers and microservices, they gain the scalability and flexibility necessary to accelerate software delivery cycles and respond more swiftly to customer needs.

With software teams now playing a critical role in achieving enterprise objectives, they need appropriate development tools, processes, and structures to succeed. This means enabling developers to work effectively—directly contributing to outcomes such as revenue generation—and efficiently, maximizing productivity and minimizing cost and risk.

For enterprise-sized teams, scaling developer solutions effectively can be challenging. While smaller developer teams can work efficiently, scaling introduces multiple bottlenecks, all of which directly impact both effectiveness and efficiency:

- Developers can become less efficient and productive as they spend increasing time configuring and managing their own software tools. This lengthens cycle times and delays delivery, increasing costs.
- Team leaders frequently struggle with limited visibility into development workflows, leading to inconsistencies in configurations, automation processes, and policy compliance. This fragmentation not only increases the risk of errors and security vulnerabilities but also complicates governance and reduces overall efficiency.
- Security teams face difficulties in assessing and mitigating risks across both workflows and software outputs, further elevating risk and impacting compliance.

Software teams use a variety of tools across development, testing, and deployment. While single-user versions of these tools enable developers to get going quickly, they often lack the advanced capabilities required to address broader organizational challenges.

In this Buyers Checklist, we focus on the container development environment that lies at the center of developer activity. We consider the critical features it must offer beyond core capabilities to overcome the challenges described above and deliver on the needs of the business. These features include:

- Centralized provisioning, configuration, and build automation, to minimize self-configuration and drive productivity.
- Security and governance across both development workflows and applications in development.

By providing centralized tooling and security governance, these capabilities empower development teams to streamline workflows, minimize manual configuration, and ensure alignment with organizational security policies. They enable developers to focus on innovation, and ensure managers and security teams maintain control of cost and risk.

This Buyers Checklist is aimed at VPs of engineering, development team leaders, decision-makers, and in the Fed Space, program managers looking to upscale their container development environments. It provides guidance on how to drive productivity, security, and manageability, ultimately reducing cost of ownership and maximizing return on investment.

2. Assessment Criteria

A container development environment facilitates the creation, selection, build, execution, and orchestration of container images and their interconnections while providing developers with a runtime view. The container development environment works alongside integrated development environments (IDEs), code repositories, testing, deployment, analytics, and other pipeline tools to create the overall development toolchain.

Below we consider:

- **Table stakes:** Core capabilities that every container development environment should provide. These do not differentiate solutions.
- **Key features:** Functional requirements that differentiate solutions from multiple vendors in this space.
- **Business criteria:** Nonfunctional requirements that technology buyers can use to evaluate and compare offerings against an organization's needs.

Table Stakes

A container development environment should provide the following core capabilities as entry point requirements. The capabilities reflect the day-to-day needs of developers but do not extend to management or security roles.

- **Configurable dashboard:** A user-friendly interface showing available container images, current execution status, and other information (e.g., security profiling).
- **Container lifecycle management:** Tools to create containers from container images, then start, stop, and monitor live containers.
- **Container build and orchestration:** Support for building container images and orchestrating multiple containers, with potential integration for Kubernetes.
- **Container networking:** It should facilitate communication between locally running containers and with hosts—for example, via port mapping or proxy HTTP/S.
- **Image library:** Access to container images from local registries, third-party tools (such as JFrog Artifactory), and public repositories.
- **Audit logging:** Capture usage information and logs for management reports and to feed into third-party tools (such as Grafana).

Key Features

More advanced, differentiating features of container development environments work toward the leadership goals of manageability and security. These include the following:

- **User access management:** Provision and deprovision accounts, and configure settings by user, role, or team. This includes file shares, integrations, and beta features, as well as access to registries and images. Integrate with common identity and access management platforms, such as Microsoft Entra (Azure's Active Directory offering) or Okta.
- **Container security:** Secure containers out of the box and reduce attack surface using features such as container network security, least privilege access to host resources, and data loss prevention.
- **Secure software supply chain:** Offer static container image composition analysis to report against common vulnerabilities and exposures (CVEs) and to enable the creation of a software bill of materials (SBOM) describing the components making up the image.
- **Secrets management:** Support the secure creation and management of passwords, tokens, and other secrets. Include capabilities to enable secrets currently in use to be reviewed against best practice.
- **Extensions management:** Control third-party extensions by enabling administrators to evaluate, curate, and provide these as a library for developers to use based on role.
- **Policy-based reporting:** Create customizable reports about how the environment is being used, for compliance/governance reasons, to identify divergence from security or corporate policies, or to feed into software development analytics (SDA).

Business Criteria

These criteria help enterprise technology buyers assess container development environments by evaluating how features like automated container provisioning, security integrations, and centralized management contribute to improved team productivity, reduced operational risks, and a more cost-efficient development pipeline. The business criteria are:

- **Ease of use:** The solution should ensure users (across developers, managers, and security teams) derive benefits with a minimal learning curve across deployment, configuration, and use. The setup should be simple and intuitive, with options like templates or self-service customization. Offer online and direct support and services to accelerate adoption.
- **Flexibility:** It should adapt to different use cases and scenarios, allowing users to use the tools, integrations, and customizations they prefer without modifying core functionality or breaching policy.

- **Security:** It is essential to mitigate security risks across development workflows, users, containers, and target environments. The solution should ensure sensitive data and critical systems are adequately protected when using the environment while remaining compliant with evolving regulations such as GDPR, SOC 2, and ISO standards.
- **Interoperability:** The solution must integrate smoothly with common repository types, CI/CD toolchains, testing automation, and related tools, such as backup/restore. Organizations should be able to work with tools, scripting languages, and target environments of their choice and need.
- **Scalability:** The environment should scale dynamically from individual users to large teams, accommodating additional complexity without escalating cost. It must scale to hundreds of developers across development and build while maintaining centralized control, auditing, and reporting.

Table 1 shows how these criteria map onto three key business outcomes, which are critical value drivers for any organization. The outcomes are: increased productivity across development teams and reduced cost and risk in the development process. By understanding which business criteria most impact these outcomes, IT decision-makers can better focus their priorities.

Table 1. Impact of Business Criteria on Business Outcomes

	INCREASED PRODUCTIVITY	REDUCED COST	REDUCED RISK
EASE OF USE	+++	+++	++
FLEXIBILITY	+++	++	+
SECURITY	+	++	+++
INTEROPERABILITY	+++	++	++
SCALABILITY	+++	+++	++

Source: GigaOm 2024

Key: + Limited Impact | ++ Moderate Impact | +++ Significant Impact

3. Docker Assessment

Docker provides a suite of solutions that accelerates the development, deployment, and management of containers by development teams. In this section, we consider how Docker's portfolio meets assessment criteria for container development environments across table stakes, key features, and business criteria. Relevant tools and solutions in the Docker portfolio are:

- **Docker Desktop:** A comprehensive container development environment platform available in paid tiers—Docker Business and Docker Teams for enterprise and team use, respectively. There are also single-user versions of Docker Desktop.
- **Docker Scout:** A security analysis tool that generates reports on and provides ongoing monitoring of software composition and risk.
- **Docker Build Cloud:** A cloud-based image build service that development teams can use collaboratively, reducing build times and costs across the team.
- **Docker Hub:** A public image repository holding Docker official, commercial/certified, and community-based container images. Within Docker Hub, Trusted Content collates a set of images created following industry best practices.

Let us now look at how these elements of the Docker portfolio relate to the key features and business criteria defined earlier in the previous section of this report.

Key Features Evaluation

Here, we consider how the Docker portfolio responds to the key features defined earlier in the report:

- **User access management:** Docker provides a console that allows administrators to configure user and role-based access to configuration features within the desktop. It offers System for Cross-Domain Identity Management (SCIM), which enables Docker accounts to be provisioned via identity providers such as Okta. The Docker Business tier supports image access management for Docker Hub—enabling access to verified rather than community-based images. Docker Business and Teams tiers also enable users to be added in bulk.
- **Container security:** Docker Desktop offers several capabilities to enhance container security within and across container architectures. Network access rules can restrict containers to specific network resources, such as internal data or test environments. A container isolation feature isolates containers from each other to prevent breaking between containers or escaping to the hypervisor, while containers can be run in a least-privilege "rootless" mode to prevent unrestricted access to compute resources.

- **Secure software supply chain:** Docker Scout provides a detailed, up-to-date listing of common vulnerabilities and exposures (CVEs) across all tiers of Docker Desktop, prioritizing issues based on severity. For all CVEs, it offers remediation recommendations for base image updates. Docker Scout also generates software bill of materials (SBOM) attestations, detailing the image composition.
- **Secrets management:** Docker Compose enables secrets to be created and used across containers. This feature is available across all Docker tiers.
- **Extensions management:** Docker extensions management is currently offered as an early access feature because it is still in development and may undergo changes based on user feedback. The Docker Business tier includes a private extensions marketplace that allows administrators to curate a set of approved third-party tools and integrations, ensuring that developers have access to vetted, secure extensions.
- **Policy-based reporting:** Docker offers audit logs and integrates with third-party solutions such as Splunk, Elk, and Grafana for visualizations. An early access feature in the business tier is the ability to review Docker usage against software supply chain best practices. Future enhancements also include SDA insights to drive best practices for developers and teams.

Business Criteria Evaluation

In terms of the defined business criteria, the Docker solution set breaks out as follows:

- **Ease of use:** Docker offers solid guidance and tutorials to those new to container-based development. While Docker Desktop's user interface is intuitive and designed for ease of use, it assumes an understanding of containers and development workflows. For novices, Docker provides extensive tutorials and documentation to bridge this gap, though more advanced functionality may require familiarity with container orchestration and pipeline management.
- **Flexibility:** Docker does its job well—that is, container-based application development. It incorporates VDI support for developers working in virtualized environments. Docker does not currently cater to mobile use cases. It does support edge and AI workloads in certain scenarios.
- **Security:** All Docker tiers offer developer security features, such as two-factor authentication. However, lower tiers tend to focus on single users or repositories. As described above, Docker Business brings capabilities to address broader security challenges plus tools for managers and security teams to manage risk across engineering teams.

- **Interoperability:** All tiers offer webhooks for integration with other platforms; integrations with third-party image registries are available to all users, with some features requiring a paid tier. Docker Scout integrations are also a cost item. Docker was a founding member of the Open Container Initiative (OCI) and fully supports the OCI specifications.
- **Scalability:** Docker Business and Teams tiers offer scalability features, such as concurrent builds, with management console features that improve scalability for fluctuating workloads. Docker Build Cloud offers increased build capacity for expanding teams, together with collaborative build features. Centralized management features reduce the time developers spend managing environments, streamline workflows across teams, and improve overall cost efficiency.

4. Analyst's Outlook

Overall, Docker's suite of solutions, led by the Docker Business subscription, effectively meets the assessment criteria outlined in this report, delivering on the benefits of increasing productivity, reducing cost and risk when developing container-based applications at scale. Docker interoperates with external repositories and tooling for software testing, data management, advanced security, and more to meet the organization's broader development needs.

Docker's suite offers significant advantages over entry-level container environments, especially in areas such as centralized security management, role-based access controls, and out-of-the-box integration with popular development tools. These enhancements not only simplify management but also reduce the operational burden on development teams, enabling them to focus on coding rather than environment setup and maintenance. We expect Docker to build out new capabilities to its portfolio over time.

It is crucial for buyers to build a clear picture of their investments in development tooling and the associated benefits, costs, and risks. Organizations already using single-user Docker tiers would be prudent to review their broader requirements, particularly where inefficiencies and unnecessary overhead may arise from the current setup.

We recommend a needs analysis across developer, manager, and security roles, together with approximate costs in terms of productivity and risk mitigation. Buyers can use the checklist below to structure this analysis, highlighting which capabilities are currently covered and which need further support.

5. Buyers Checklist

Use the criteria described in this report to determine your organization's needs by considering whether a certain option is a top or secondary priority. You may need to consider the needs of different stakeholder groups, in which case, you can use this checklist as a workshop tool.

You can then compare your priorities against vendor capabilities to determine the best value fit for your organization.

GIGAOM

BUYERS CHECKLIST

KEY FEATURES	TOP PRIORITY	SECONDARY PRIORITY	NOT A PRIORITY
USER ACCESS MANAGEMENT			
CONTAINER SECURITY			
SECURE SOFTWARE SUPPLY CHAIN			
SECRETS MANAGEMENT			
EXTENSIONS MANAGEMENT			
POLICY-BASED REPORTING			

BUSINESS CRITERIA	TOP PRIORITY	SECONDARY PRIORITY	NOT A PRIORITY
EASE OF USE			
FLEXIBILITY			
SECURITY			
INTEROPERABILITY			
SCALABILITY			

Source: GigaOm 2024

6. About Jon Collins

Jon Collins has over 35 years of experience in IT. He has worked as an industry analyst for a number of years and has advised some of the world's largest technology companies, including Cisco, EMC, IBM, and Microsoft in product and go-to-market strategy. He has acted as an agile software consultant to a variety of enterprise organizations, advised government departments on IT security and network management, led the development of a mobile healthcare app, and successfully managed a rapidly expanding enterprise IT environment. Jon is frequently called on to offer direct and practical advice to support IT and digital transformation initiatives, has served on the editorial board for the BearingPoint Institute thought leadership program, and is currently a columnist for IDG Connect.

Jon wrote the British Computer Society's handbook for security architects and co-authored The Technology Garden, a book offering CIOs clear advice on the principles of sustainable IT delivery.

7. About GigaOm

GigaOm provides technical, operational, and business advice for IT's strategic digital enterprise and business initiatives. Enterprise business leaders, CIOs, and technology organizations partner with GigaOm for practical, actionable, strategic, and visionary advice for modernizing and transforming their business. GigaOm's advice empowers enterprises to successfully compete in an increasingly complicated business atmosphere that requires a solid understanding of constantly changing customer demands.

GigaOm works directly with enterprises both inside and outside of the IT organization to apply proven research and methodologies designed to avoid pitfalls and roadblocks while balancing risk and innovation. Research methodologies include but are not limited to adoption and benchmarking surveys, use cases, interviews, ROI/TCO, market landscapes, strategic trends, and technical benchmarks. Our analysts possess 20+ years of experience advising a spectrum of clients from early adopters to mainstream enterprises.

GigaOm's perspective is that of the unbiased enterprise practitioner. Through this perspective, GigaOm connects with engaged and loyal subscribers on a deep and meaningful level.

8. Copyright