# GigaOm Radar for Enterprise CI/CD

## TABLE OF CONTENTS

1. Summary
2. About the GigaOm Radar
3. Market Categories & Deployment Types
4. Key Criteria Comparison
5. GigaOm Radar
6. Vendor Roundup/Overview
7. Conclusion
8. About Jon Collins
9. About GigaOm
10. Copyright
1. Summary

The goal of an end-to-end Continuous Integration/Continuous Delivery (CI/CD) pipeline is to deliver software-based innovation and business value at both speed and scale. As we saw in the Key Criteria Report: Enterprise CI/CD report, continuous principles apply across the pipeline to integration, delivery, and deployment into test, pre-production, and production environments. An important concept is the ability to represent everything in the pipeline, and the pipeline itself, as inter-dependent artifacts (Figure 1).

**FIGURE 1. BLUEPRINT/ARCHITECTURE OF THE END-TO-END CI/CD PIPELINE**

The CI/CD pipeline should align with broader elements of innovation best practice, each of which needs to incorporate the ‘continuous’ principle including:

- Communication and collaboration
- Governance, compliance and risk
- Management and planning
- Evaluation, visibility and feedback

Together, these concepts make up the framework for how we can consider the shape of CI/CD today, and offer criteria for how enterprises can evaluate vendors of CI/CD tooling. In this report we provide a broad overview of the tools vendors are offering in response.
2. About the GigaOm Radar

HOW TO READ THE GIGAOM RADAR

The Market Radar is part of a series of documents aimed at giving the reader the tools to better understand a technology, evaluate it, and explore the market to find the best solutions for their organization.

In this context, and to get a complete view of the state of the solutions available in the market, the reader should consider the following documents:

Key Criteria is an introduction to the technology, defines the necessary evaluation metrics, the key criteria to evaluate new solutions, and the impact of the latter on the former. It is dedicated to those end users that are approaching a new technology for the first time or want an update on the latest evolution.

Market Radar offers a brief 360° view of the market, including market and technical positioning of most notable vendors, and a short introduction to their solutions and differentiation, including a high-level graphic comparison of the vendors.

Vendor Profile covers a single vendor regarding the solutions described in the other reports. It provides more details on the solution, how the vendor approached the key criteria, and the impact that its solution has on the evaluation metrics. This document helps end-users to get a quick but complete evaluation of a single vendor.
3. Market Categories & Deployment Types

We can differentiate three groups of CI/CD vendors, which together respond to the need to demonstrate continuous value delivery across the software development pipeline:

- **Pipeline providers** who offer tooling and integration points to automate the entire software development workflow

- **Visibility vendors** whose tooling-agnostic interest is in offering a coherent view of the pipeline, its efficiency, and effectiveness

- **Deep experts** who focus their attention on handling the evolving complexities of integration and deployment

For a better understanding of the market and vendor positioning, we categorized solutions for CI/CD by target market segment as follows.

- **Small-medium enterprise solutions**: This category covers solutions which are more fitting for smaller organizations, which means they are simple to deploy and get started.

- **Large enterprise solutions**: These solutions have more extensive features and better scale, yet may have a greater learning curve than simpler solutions.

We have also considered their predominant deployment models:
• **On-premises:** Software deploys on an organization’s own servers, which means scaling is dictated by the resources the organization provides.

• **Cloud-hosted:** Similar to on-premises solutions, but installs in the public cloud. The user still keeps full control of the infrastructure, and therefore costs.

• **SaaS:** These solutions are entirely offered as a service, with costs based on a per-use model.

The overall goal is to give the reader a broader perspective and context regarding the different solutions available in the market and how they can fit with his organization and business needs.

### TABLE 1. VENDOR POSITIONING

<table>
<thead>
<tr>
<th>MARKET SEGMENT</th>
<th>VENDOR GROUP</th>
<th>DEPLOYMENT MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small/Medium Enterprise</td>
<td>Large Enterprise</td>
<td>Visibility</td>
</tr>
<tr>
<td>AWS</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>CircleCI</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>CloudBees</td>
<td>++</td>
<td>+++</td>
</tr>
<tr>
<td>CodeFresh</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Gitlab</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>IBM</td>
<td>++</td>
<td>+++</td>
</tr>
<tr>
<td>JFrog</td>
<td>++</td>
<td>+++</td>
</tr>
<tr>
<td>Micro Focus</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Microsoft</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Red Hat (dev preview)</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Xebialabs</td>
<td>++</td>
<td>+++</td>
</tr>
</tbody>
</table>

+++ = strong focus and perfect fit of the solution
++ = The solution is good in this area, but there is still room for improvement
+ = The solution has limitations and a narrow set of use cases
- = Not applicable or absent

Source: Gigaom 2020
4. Key Criteria Comparison

Following the general indications introduced with the Key Criteria Enterprise CI/CD Report, table 2 quickly summarizes how each vendor included in this research performs in the areas that we consider differentiating and critical for CI/CD pipelines. The objective is to give the reader a snapshot of the technical capabilities of different solutions and define the perimeter of the market landscape.

### TABLE 2. KEY CRITERIA AND EVALUATION METRICS COMPARISON

<table>
<thead>
<tr>
<th>KEY CRITERIA</th>
<th>EVALUATION METRICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline Automation</td>
<td>Feature Set</td>
</tr>
<tr>
<td>Environment Management</td>
<td>Scalability</td>
</tr>
<tr>
<td>Infrastructure Orchestrating</td>
<td>Flexibility and Usability</td>
</tr>
<tr>
<td>Pipeline Measurability</td>
<td>Solution Ecosystem</td>
</tr>
<tr>
<td>Everything As Code</td>
<td>Licensing and Support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Pipeline Automation</th>
<th>Environment Management</th>
<th>Infrastructure Orchestrating</th>
<th>Pipeline Measurability</th>
<th>Everything As Code</th>
<th>Identity-based Access</th>
<th>Feature Set</th>
<th>Scalability</th>
<th>Flexibility and Usability</th>
<th>Solution Ecosystem</th>
<th>Licensing and Support</th>
<th>Overall ROI/TCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>CircleCI</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
</tr>
<tr>
<td>CloudBees</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>CodeFresh</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>GitLab</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>IBM</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>JFrog</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>Micro Focus</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td>Microsoft</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td>Red Hat</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>XebiaLabs</td>
<td>+++</td>
<td>+</td>
<td>-</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+++</td>
</tr>
</tbody>
</table>

*+++ = strong focus and perfect fit of the solution
+++ = The solution is good in this area, but there is still room for improvement
+ = The solution has limitations and a narrow set of use cases
- = Not applicable or absent.

Source: GigaOm 2020

By combining the information provided by table 1 and table 2 the reader should be able to get a clear idea of the market and technical solutions available in the market.
5. GigaOm Radar

All the Key Criteria and the critical feature impact analysis are consolidated in the following graphic representation: The GigaOm Radar (figure 3). This vector-based graphic gives an overall perspective on all the vendors included in this research in terms of technical capabilities and features (table 2), and execution on the vision, regardless of their market share or segment (table 1). The GigaOm Radar focuses on technical capabilities, roadmap, execution, and the ability to innovate. In this graphic, each vendor representation of the market, there are three fundamental informational points:

- **The current position** in the chart gives information on the state of the solutions provided by the vendor
- **The direction** indicates where the vendor is going with the product strategy and development
- **The module** signifies how fast it is executing on vision and strategy

The metrics on which the GigaOm Radar focuses:

- **Maturity:** Expresses the maturity and solidity of the product, user acceptance of the solution, and overall ecosystem sustainability.
- **Innovation:** The differentiation of the solution, innovative technical aspects, and overall vendor approach to the market.
- **Feature Play:** Gives indications about the focus of the solution in terms of single differentiating features and technical aspects of the product. It is usually including new vendors that are defining new product categories, niche players, and point solutions.
- **Platform Play:** contrary to “Feature Play,” solutions positioned in this area provide a horizontal platform that can face a broader range of challenges, with a comprehensive feature set and an extensive ecosystem.

The GigaOm Radar is organized in three concentric areas and, the closer to the center, the better:

- **Leaders:** All the vendors that are competing for market leadership for the metrics described above, even if they are competing in different market segments.
- **Challengers:** All the vendors with the potential to become a leader soon, niche, or traditional players with an established market, and other companies that have interesting solutions but still have to prove their real differentiation.
- **New entrants:** usually companies with a limited feature set and too young to be included in inner areas, but with potential to move there soon.
- The central area of the GigaOm Radar is usually empty. It is reserved for extremely mature and consolidated markets with very few vendors remaining and solutions that are mature, comparable,
and without much space for further innovation.

The market share metric is excluded from the GigaOm Radar to give a more clear view of the potential of the vendors. It is a forward-looking market landscape representation that leaves more room for innovation and differentiation instead of weighting established positions.
6. Vendor Roundup/Overview

**AWS**

AWS has built a portfolio of tools covering the full CI/CD toolchain, including AWS CodeStar as an end-to-end customizable pipeline for AWS. The portfolio also incorporates AWS CodePipeline, which enables pipelines to be configured and used as per team needs, incorporating automated or manual approval gates. The provider also works with a broad ecosystem of tools vendors. AWS enables automated deployment to its AWS infrastructure and services (such as EC2, Fargate, or Lambda), a customer’s on-premises servers, or other platforms that support its CodeDeploy tool. Meanwhile, AWS Cloud Development Kit (CDK) enables target infrastructure to be documented ‘as code’ alongside application source, working with AWS CloudFormation and AWS CodeDeploy. Overall, the company’s developer focus and very low-cost base, make its CI/CD pipeline tooling highly attractive for AWS shops in particular, across on-premise and hybrid models. AWS delivers what its customer organizations need in terms of automated, continuous integration and delivery, in a secure, well-governed environment that supports public-private hybrid use cases. **Strengths:** Provides a comprehensive platform for AWS customers in particular. One to watch as it increases the use of machine learning (ML) across the development lifecycle (via the CodeGuru brand). Tool costs are negligible, e.g., AWS CodeStar is free; AWS CodePipeline costs $1 per active pipeline, with no price constraint on parallel actions. **Weaknesses:** The toolchain largely aims at AWS or on-premise targets.

**CircleCI**

As a ‘deep expert’ CI/CD vendor, CircleCI aims to simplify what can be a complex space, in terms of both artifacts and activities involved. And, to a large extent, it achieves this. The CircleCI solution uses Yet Another Markup Language (YAML) to represent build steps and deployment targets, workflow targets, and its own configuration. This means that anything it uses can be stored under configuration control. YAML templates, called Orbs, can be used to standardize steps, deployment targets, and integrate third-party pipeline tools. The product comes with an amount of ‘opinionation’ (i.e. in-built best practice), supporting the notion of guardrails without enforcing too much control. Workflows can be scheduled and can incorporate tags/regular expressions for more complex build scenarios, as well as manual review steps, should a build require approval before deployment, for example. CircleCI looks to address organizations that no longer find traditional, legacy CI/CD (e.g., Jenkins) infrastructure suitable, as well as helping cloud-native organizations and projects adopt forward-looking deployment targets, for example, containers and microservices. It can be installed on a private server, or run from the public cloud, on a pay-per-use basis. **Strengths:** CircleCI focuses unapologetically on addressing ‘core’ CI/CD functionality, which differentiates from broader, yet less deep solutions. It integrates with a broad gamut of platforms and tools, with some 900 orbs in its registry, from 45 partners. **Weaknesses:** Identity-based access could be broader in terms of user types but covers the main bases. Some customers see cost control as a potential challenge, but this is in part due to the scale and complexity of the applications created.
CloudBees

CloudBees is a company in transition, with what can be seen as two, historically independent end-to-end offerings. First, the company is a key provider of the extensively used open-source Jenkins CI tool (which it wraps into its CloudBees Core product). Acquisitions of two CD companies (notably Electric Cloud) gave the company an end-to-end capability that included release and deployment, with traditional CI for legacy and modern cloud/cloud-native workloads, and dashboard-based value stream analytics and insights across the pipeline. The company also founded Jenkins X, a serverless CI/CD platform based on open source Tekton CI/CD tasks/pipelines. Jenkins X gears itself around the needs of cloud-native organizations looking to deliver into container-based/Kubernetes environments. Software as a Service (SaaS)-based Jenkins X is, in principle, serverless, so only runs on demand, and is entirely based on YAML. It is also strongly opinionated in terms of the toolchain, which is a bonus for efficiency and control: it was built for GitOps best practice, for example. Overall, CloudBees offers a flexible, feature-rich, and comprehensive portfolio that will integrate increasingly over time. The company is working to ensure that it evolves to meet the changing needs of its clients (noting that organizations and departments move at different speeds), and the broader industry. Its direction is couched in terms of what it calls Software Delivery Management (cross-pipeline visibility and governance) and Software Delivery Automation (pipeline automation, based on open source or commercial offerings from CloudBees and other vendors). SDM and SDA provide a unified platform to support integrated solutions across both traditional CI/CD and faster-changing, future-facing environments. **Strengths:** Powerful, extensible, and flexible portfolio delivering on traditional and cloud-native needs at an enterprise scale. CI/CD engine-agnostic approach is attractive for existing customers looking to evolve. **Weaknesses:** Traditional Jenkins is sometimes seen as cumbersome; its historical extensibility has resulted in over-complex pipelines. Jenkins X is geared towards Kubernetes and works with git-based repositories only.

Codefresh

Codefresh offers an end-to-end pipeline capability. It is Kubernetes-native, applying container-based principles to the actions of CI/CD. Each activity takes place in its own container, enabling reuse and meaning that off-the-shelf, internally scripted, or third party steps (e.g., running a vulnerability scan) are treated in the same way. Codefresh works in any Kubernetes installation, in-house, cloud, or indeed desktop. Codefresh majors on build speed, offering distributed caching, and parallel builds. As it is Kubernetes-native, it can spin up environments, and indeed demonstrate deployments, more quickly. In terms of differentiating features, Codefresh has embraced the notion of pipelines as code, offering pipeline breakpoints in the same way a debugger might use application breakpoints. Pipelines can be shared between teams. The tools also look to simplify the management and use of Helm packaging and deployment charts. They incorporate a dashboard for monitoring containers post-deployment, representing Kubernetes clusters as Codefresh objects. Any environment (demo, test, deployment) can be built and managed in terms of both cluster status and build dependencies. While Codefresh is open - it works with traditional applications and workloads, as well as serverless environments - it aims particularly at organizations looking to accelerate Kubernetes adoption across on-premises, hybrid, multi-cloud or edge-based deployments. It incorporates several features synonymous with cloud-native delivery, such as GitOps support, and blue-green deployments via feature flags. **Strengths:** Focused on improving build times, through techniques such as distributed image caching and use of local storage. Pipelines as code, with breakpoints for debugging. Target environments are a first-class citizen.
**Weaknesses:** Cost management can be seen as an issue for more complex build requirements. Out-of-box integration is with git-based repositories only.

**GitLab**

GitLab comes to market with a unified, end-to-end pipeline capability based around the open-source git repository. Within this portfolio, it offers GitLab CI, a general-purpose CI platform. Pipelines are based on YAML, so they can be stored alongside code. The GitLab solution is nothing if not comprehensive. GitLab includes (binary) artifact management, deployment chart registry, and application security scans within the CI pipeline; meanwhile, analytics and management information can be fed (via a ‘cycle analytics’ report) into process improvement. GitLab’s main differentiator is to offer a single, integrated application rather than a best-of-breed offering (though it does not require customers to adopt all elements of its solution). While it has not historically focused on integration with third-party applications, this is a growing area of development for GitLab: recent releases integrate better with ‘first-class’ tools such as Jira, for example. While the environment is not target-specific, the company recognizes the increasing needs of cloud-native organizations, for example, offering test environment management for Kubernetes targets. The company recently released an ‘auto-DevOps’ feature, which looks to create a Kubernetes-based CI/CD pipeline automatically, based on project setup criteria. GitLab is available as both SaaS and self-hosted, with the same features available at the same price levels. **Strengths:** Single source of tooling increases pipeline coherence and reduces fragmentation. Good for building complex pipelines. GitLab can scale build resources up and down to keep build costs under control. **Weaknesses:** GitLab is git-only, and (by nature) does not major on integration with other tools.

**IBM**

IBM’s decision to embrace open source in development and operational environments has set the scene for its CI/CD automation approach. The company’s Cloud toolchain approach incorporates the Eclipse Orion Web IDE, Git Repositories, Issue Tracking, and a Delivery Pipeline. Which, in turn, works with all the expected open-source and third-party DevOps tools: IBM’s DevOps platform capabilities made up of Urbancode and Rational Test. The IBM approach is to see the toolchain as a first-class citizen. The company is largely agnostic about the tools to choose from. However, it offers a variety of templates for common toolchains or target environments, and a core DevOps platform to manage governance and quality across the toolchain. The notion of templates overlaps with IBM Cloud Paks - these are self-service environments oriented towards specific environments such as Kubernetes but also legacy environments. Understandably, IBM’s capabilities increasingly run on Red Hat tooling and capabilities, particularly OpenShift. As illustrated by IBM’s Cloud Paks and DevOps for CloudPaks offerings, IBM offers an attractive option for organizations with a large amount of legacy development and delivery tools, which can either be enhanced or modernized both as a toolchain or as a target environment. **Strengths:** IBM offers a full suite of on-premises, cloud, and hybrid cloud DevOps tools, as well as broad support for a wide variety of external tools for an end to end toolchain. IBM brings cognitive analytics into the mix, from standard insights about delivery speed to diagnosis of the potential risks. **Weaknesses:** Start-ups might not see IBM as an obvious partner, though the vendor should not be ruled out, particularly as it sees the value of optimizing the end to end pipeline through development, operations, and back.
JFrog

JFrog’s focus is on the management and security/compliance of the artifacts created by CI (through its Artifactory and Xray products respectively) and of the pipeline stages that surround them (via Pipelines, previously Shippable). Together, these capabilities make up the integrated JFrog Platform. This is agnostic towards other tools in the pipeline: while it includes CI, it can work with other CI tools and repositories; it is also agnostic to the artifacts (e.g., binaries, container images) it manages. JFrog Pipelines developed with containers in general, and Kubernetes, in particular, in mind. Pipelines are declarative (rather than scripted) and can exist as sub-pipelines to incorporate into multiple other pipelines - a ‘pipeline of pipelines’ approach. The JFrog Platform incorporates data collection across all services, tools, pipeline steps, artifacts, permissions, and security data, enabling users and managers to track the health of their releases and optimize their processes. JFrog tends to work with larger organizations that have historically followed manual processes and are now looking to relieve this burden while reducing risk by adding security, repeatability, and traceability. JFrog’s approach leads with artifact management rather than process, suiting organizations looking to start simple and then expand. JFrog products can be deployed on-premises, in the cloud, or a hybrid model. In addition, the vendor offers SaaS subscription plans on AWS, GCP, and Azure platforms. **Strengths:** The integrated JFrog platform offers much of what organizations need in terms of managing both process and delivery. A major plus is that security scanning is seen as an inherent part of the pipeline, offering DevSecOps out of the box. **Weaknesses:** Costs for SaaS offering depend on the level of use (based on throughput and storage usage), so these need to be monitored. While it can work for other target environments, JFrog Pipelines is designed for Kubernetes targets. Current tools do not support the concept of environment state, so environments cannot be managed as ‘first-class citizens’.

Micro Focus

Micro Focus offers Pulse CI within its Dimensions CM configuration management tool, as well as supporting other Micro Focus and native Git systems. Micro Focus Release Control then offers target-agnostic integration and visibility of build objects across development, deployment, and operations. Micro Focus uses the term ‘expert chain’ to represent a series of steps/plugins that define the flow of continuous inspection and build review and approval, including what third-party tools are used, by which people/roles. Pulse CI supports code inspection and review as enforceable quality gates. The solution enables architects to create and manage pre- and post-deployment environments, as logical (Terraform-based) definitions mapping onto physical servers. These can be version controlled, as can anything that exists as a file, including environment/target and process definitions. For example, processes such as “deploy app” can be exported in XML or JSON and then managed as code. Micro Focus mainly targets its substantial enterprise customer base, but the product set would be suitable for the broader, cloud-native audience, particularly as they outgrow fragmented point-product pipeline approaches. **Strengths:** Micro Focus has created a solid offering for end-to-end CI/CD, enabling customers to build well-governed, flexible, and manageable pipelines. **Weaknesses:** Pulse CI is available within a broader toolset (e.g., Dimensions CM) rather than as a product in its own right. A strange weakness is that the company could do more in promoting a solid (albeit relatively new) CI/CD competence, both to its customers and internally.
Microsoft

Microsoft’s core product for CI/CD is Azure Pipelines. Contrary to what its name suggests, it is not specific to the Azure target; it also supports Windows, Mac, and Linux environments, and works with both GitLab and Atlassian Bitbucket as well as Microsoft repositories (including Github). It has built-in support for Docker / Kubernetes and works with virtual machine (VM) managers to enable deployment to any VM environment. Pipelines can be simple or complex based on role-based access and quality gates, all of which can be modeled, managed, and monitored. Build jobs can run in parallel, speeding up integration and delivery times. The pipelines offering also supports YAML defined pipelines, enabling pipelines to be stored with code. While Azure Pipelines will be the obvious choice for Microsoft shops looking to streamline their development workflows, the company is targeting the broader cloud native community, at the same time as offering a portal onto existing/legacy tools such as Jenkins. Open source projects can use Azure Pipelines for free, with up to 10 parallel jobs. More broadly, the tools align with Microsoft’s efforts to automate and simplify the development of applications with a machine learning element: cf our report on MLOPs [title/reference]. **Strengths:** Azure Pipelines is well-suited to smaller teams/projects. The ensemble works well together, benefiting from facilities such as identity management to enable role-based access to specific services. **Weaknesses:** While Microsoft has done its best to minimize fears of lock-in to the platform, some remain. There is more that could be done to increase flexibility (including caching), automation, and insight across its own environment.

Red Hat (developer preview)

Red Hat offers a developer preview release of OpenShift Pipelines for CI/CD. The preview is a Kubernetes-native CI solution based on the open-source Tekton project (see also CloudBees/Jenkins X above). It runs serverless, which means it only consumes resources when in use. As it is container-based, developers can work in an environment that is identical to production. Meanwhile, Red Hat Ansible supports packaging and (continuous) deployment automation via ‘playbooks,’ enabling application environments to be spun-up and tested as-code, both in terms of infrastructure and connectivity. Red Hat also supports Helm charts; as both formats are stored as code. Red Hat advocates a GitOps approach in which the Git repository is used as a single source of truth. The resulting combination incorporates cloud-native capabilities such as support for canary deployments and feature flagging. It also has a focus on simplifying and automating deployment, for example, enabling roll-back to a last known good configuration. While the company is heavily invested in Kubernetes, it is not target-specific: it currently hooks into Jenkins for ‘traditional’ CI/CD implementations. As OpenShift Pipelines for CI/CD is one element of OpenShift, it is most likely to be of interest to existing customers who are looking to move towards more future-facing, cloud-native architectures. As per the Red Hat model, it is also appropriate for organizations looking to avoid lock-in to a cloud vendor, via hybrid or multi-cloud approaches. Pricing for Red Hat software is on a yearly license basis. **Strengths:** Future-facing environment. Focus on hand-off between development and operations. This works well for multi- and hybrid cloud deployments. No cost limitations on builds or functionality. Can run standalone. **Weaknesses:** Developer preview only, i.e., not for production or business-critical workloads at this point. More geared around cloud-native environments. While built on Tekton, OpenShift Pipelines for CI/CD is only available (from Red Hat) as part of a broader license. More geared around integration/delivery than an end-to-end pipeline.
XebiaLabs offers a management and control layer which operates across CI/CD, as well as other pipeline stages. The provider connects a broad range of DevOps tools, and deploys to a diverse set of environments, from legacy to Kubernetes. The former it accomplishes through APIs and the latter via an as-code approach, which pervades every aspect of the XebiaLabs solution. For example, the XL Deploy deployment engine enables target environments to be modeled and managed either ‘as code’ or using a graphical tool. The XebiaLabs model is role-based, with permissions allocated to pipeline steps according to users or groups. It incorporates the notion of ‘patterns,’ i.e., pipeline and deployment templates, which can be shared between and across teams. The solution pulls data from across the pipeline into a central repository. Reporting dashboards enable stakeholders to view analytics according to their needs, within or outside development. XebiaLabs targets the needs of organizations looking to scale their efforts or deal with complexity, e.g., through sharing pipelines across teams. The solution is a good option for organizations looking to bring their CI/CD practices (back) under control and release valuable engineering resources. The XebiaLabs DevOps Platform is ostensibly a server-based solution, though it can be installed to an AWS instance. **Strengths:** Standardization by way of templated pipelines, enabling automation in development, and across to deployment. The breadth of tools it can integrate via API connections, covering the entire software delivery process. XebiaLabs goes further than some other tools in terms of (predictive) analytics. **Weaknesses:** Simpler environments may well be comfortable still with core solutions or do-it-yourself toolchains. XebiaLabs does not offer features to enable build efficiency to be controlled (e.g., through build optimization, parallel builds, etc.).

XebiaLabs is merging with CollabNet VersionOne: this could affect future plans.
7. Conclusion

Each of the vendors in the list brings something different to the DevOps pipeline in general, and CI/CD in particular. For example, for very different reasons, AWS, Red Hat and Micro Focus are more likely to be used by existing customers; CircleCI and JFrog are strong in pipeline description and management; CodeFresh and GitLab emphasize features to keep control of build management and costs; Red Hat, CloudBees/Jenkins X, and Jfrog are geared towards Kubernetes. Clearly, there is no such thing as a ‘generic’ CI/CD solution; there is also a question of single-provider vs. best-of-breed. GitLab proudly offers the former, while Microsoft is keen to express its openness to external options, and IBM and XebiaLabs emphasize how they hook into a wealth of providers. Meanwhile, Red Hat and Micro Focus CI/CD cannot easily deploy outside of a broader product set. Cost models are also very different. AWS offers a very low cost of entry, for example, offering the tool as an enabler rather than its core value-add. Tools from smaller providers can be seen as an issue. However, for pay-as-you-go vendors, this is often down to build efficiency rather than a tool being inherently expensive (cf. CircleCI and CodeFresh). This creates an area of contention - if costs spiral out of control as engineers make too much use of a tool (for example, to test multiple scenarios), is this a good or bad thing? While it could be argued that build engineering should be kept under control, equally, we will likely see differentiating features emerge that help both visibility and build efficiency. Above all, we perhaps see a landscape in transition. On the point about build costs, vendors increasingly see build efficiency as a differentiator, for example, enabling builds to take place on under-utilized local resources. End-to-end security is also an area of differentiation. In essence, vendors such as JFrog (who were earlier to market with artifact scanning) may be ahead of the game now, but not forever. For all of these reasons, it is a case of selecting the most appropriate to fit the environment, and the challenges, at hand. For example, IBM may be an appropriate option for not cloud-native start-ups, even if it is not the obvious choice. As a final, general point, criteria such as ease of use, learning curve, performance, and long-term costs should be weighed against the more significant advantages of getting the basics right: automating CI and CD activities, reducing risk and increasing governance, and of creating an environment suited to the delivery of modern, future-facing applications. Adopting any one of these solutions will take an organization significantly further down the track of improving software delivery efficiency and effectiveness.
Jon Collins has advised the world’s largest technology companies in product and go to market strategy, 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 upon to offer direct and practical advice to support IT and digital transformation strategy 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 six principles of sustainable IT delivery. He has written innumerable papers and guides about getting the most out of technology and is an accomplished speaker, facilitator, and presenter.
9. 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.
10. Copyright

© Knowingly, Inc. 2020. "GigaOm Radar for Enterprise CI/CD" is a trademark of Knowingly, Inc.. For permission to reproduce this report, please contact sales@gigaom.com.
VENDOR PROFILE

JFrog

Key Criteria for CI/CD

JON COLLINS

TOPIC: DEVOPS
1. Summary

JFrog’s focus is on the management and security of the artifacts created by CI, and of the pipeline stages that surround them. It can be seen, therefore, as a key element of any response to the challenges caused by inefficient creation, management, and deployment of software artifacts.

Market positioning: JFrog sets its stall around removing friction from the software development and deployment process – an aspiration it calls ‘liquid software.’ The JFrog Platform offers the following capabilities:

- **Artifactory** – software artifact management
- **Pipelines** – CI/CD pipeline automation and optimization
- **Xray** – security and compliance scanning of binary artifacts and container images

While the JFrog Platform includes CI as a core capability, it can work with other CI tools (e.g. Jenkins and Circle CI). Artifactory is agnostic to the composite units of delivery (e.g. binaries, container images) it manages. It does not replace a version control tool such as Git: its focus is on managing dependencies within and between these units of delivery.

Meanwhile, Pipelines comes with workflow automation capabilities and can orchestrate the DevOps toolchain. It therefore automates the delivery process and visibility of the places where artifacts are being deployed. This requires repositories to be kept in alignment: JFrog recommends that repository
names and promotion policies mirror environments in the pipeline (e.g. development, testing, production).

A major plus in the market is that (via Xray) security scanning is seen as an inherent part of the pipeline. JFrog tends to work with larger organizations that historically have followed manual processes and are now looking to relieve this burden, while reducing risk by adding security, repeatability, and traceability.

**Deployment Model:** JFrog products can be deployed on-premise, in the cloud, or in a hybrid model. In addition, the vendor offers SaaS subscription plans on AWS, GCP and Azure platforms.
2. About the Vendor Profile

HOW TO READ THIS REPORT

The Vendor profile is part of a series of documents aimed at giving the reader the tools to better understand technology, evaluate it, and explore the market to find the best solutions for his organization.

In this context, and to get a complete view of the state of the solutions available in the market, the reader should consider the following documents:

Key Criteria to Evaluate Hybrid Cloud Data Protection is an introduction to the technology, defines the necessary evaluation metrics, the key criteria to evaluate new solutions, and the impact of the latter on the former. It is dedicated to those end-users that are approaching a new technology for the first time, or want an update on the latest evolution.

Vendor Profiles for Key Criteria are easy-to-read deep dive documents that cover a single vendor regarding the solutions described in the other reports. They provide more details on the solution, how the vendor approached the key criteria, and the impact that its solution has on the evaluation metrics. This document helps end-users to get a quick but complete evaluation of single vendors.
3. Key Criteria Analysis

**Pipeline automation** – As well as offering core CI/CD, JFrog plays a role in automating the broader delivery process to support repeatability and management of builds, enabling efficient redeployment and upgrade. This includes triggering pipeline steps based on events (such as a date or authorization).

**Environment management** – current tools do not support the notion of environment state, so these cannot be managed as ‘first-class citizens.’ However, artifacts can be labeled according to deployment target, e.g. test or production. Third-party deployment tools (e.g. ansible) can be automatically triggered from Pipeline steps.

**Pipeline measurability** – the JFrog Platform incorporates data collection across all services, tools, pipeline steps, artifacts, permissions and security data, enabling users and managers to track the health of their releases and optimize their processes. For example, users can see all pipelines as running/failed/succeeded, identify bottlenecks, or pinpoint security vulnerabilities and their “impact radius” across different releases. The vendor is looking at how that information can be used more in the future, for example, to help optimize lead times across pipelines.

**Everything as Code.** JFrog Pipelines are declarative (rather than scripted) based on YAML, and can be modularized – for example, ‘post-integration testing’ can be configured as a sub-pipeline which can be incorporated into multiple other pipelines – a so-called ‘pipeline of pipelines’ approach.

**Governance by Design** – quality enforcement gates are a standard element of the JFrog approach – not only for management and compliance but also for security. Pipelines enable users to define gates or alerts for certain tasks, as well as trigger compliance pipelines that automatically ensure the release meets regulatory and company policies. Furthermore, Xray security or drift alerts can automatically block download or deployments of compromised packages.
4. Evaluation Metrics Analysis

**Feature set** – as mentioned, JFrog’s portfolio is about artifact and process management. It offers features to enable CI/CD to happen smoothly, securely and in a well-governed way, which are both comprehensive and solid.

**Flexibility and usability** – JFrog does not look to be prescriptive in how pipelines are constructed. However, tools are opinionated in terms of quality gates and general governance which could lead to additional deployment overheads (as these need to be understood and defined). For example, Artifactory and Pipelines incorporate the notion of attestations, to assure that a given artifact has not been tampered with and to confirm its provenance (see DevSecOps report).

**Solution ecosystem** – While JFrog doesn’t see itself as an integrator, the JFrog Platform is agnostic about which other tools are in the pipeline.

**Licensing and support** – Licensing for on-premises or hybrid deployments is offered on a yearly basis according to functionality, with no restriction on number of users. Licensing for the cloud SaaS offering is on a monthly basis. 24/7 support contracts are available for higher-tier packages. The company offers an open-source version of Artifactory for management of binaries created by Maven, Gradle and Ivy, and a free Community Edition for C/C++ artifacts.

**Overall ROI/TCO** – with Artifactory at the core, the JFrog Platform is designed to overcome the challenges and costs associated with management of binaries and other packaged software components, and to automate and streamline the delivery process of continuous software updates. The company has a low entry costs: while costs are dependent on level of use, users can maintain visibility of cost-effectiveness as they go.
5. Future-Facing Features

Analytics insights – JFrog is looking to further leverage the (meta)data it has been generating from across the development pipeline. This includes use of ML for more predictive insights and pipeline optimization, so the vendor will be one to watch across the coming year.

Microservices/Kubernetes support – As already described, JFrog Pipelines was developed with containers in general, and Kubernetes in particular, in mind. The company is already geared towards forward-looking microservices/Kubernetes environments, and is enhancing its environment management offerings to support the complexity involved.
6. Bottom Line

Potential customers can see the JFrog Platform as emerging from binary management to delivering an end-to-end, unified, DevOps platform – encompassing artifact management, pipelines, and security built into the pipeline. Nonetheless, JFrog’s approach still leads with artifact management rather than process, suiting organizations looking to start simply and then expand (given how artifacts are less likely to differ massively, even if multiple pipelines exist). This also enables a compliance-first approach based on controlling artifacts rather than restricting process stages.

JFrog therefore becomes a clear candidate for organizations struggling with artifact management (which may not always be the case, for example, if already locked into a specific, simple delivery pipeline and toolchain) or looking to integrate security into their delivery. Meanwhile, decision makers need to recognize that there are no magic bullets: perhaps it is process or architecture issues causing the challenges, which should be solved holistically rather than relying on a tool alone.

Note that Pipelines is not restricted to ‘cloud-native’ deployments based on (say) containers or Kubernetes, however it has been developed with these in mind. While it can be used to replace legacy pipeline tools, it’s more likely that an organization would migrate to a JFrog-based environment in parallel with modernizing its development practices.
7. About Jon Collins

Jon Collins has advised the world’s largest technology companies in product and go to market strategy, 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 upon to offer direct and practical advice to support IT and digital transformation strategy 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 six principles of sustainable IT delivery. He has written innumerable papers and guides about getting the most out of technology and is an accomplished speaker, facilitator, and presenter.
8. 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.
9. Copyright

© Knowingly, Inc. 2020. "JFrog" is a trademark of Knowingly, Inc. For permission to reproduce this report, please contact sales@gigaom.com.