



Modernize Applications with Open Source Software on Google Cloud

RESEARCH BY:



Dave McCarthy
Research Vice President,
Cloud and Edge Infrastructure Services, IDC



Table of Contents

Click on titles or page numbers to navigate to each section.

IDC Opinion	3
Open Source Software on Cloud Is Accelerating Enterprise Application Modernization	4
Notes from the Field	10
Investment Services Firm Accelerates Legacy Infrastructure Transformation by Leveraging Open Source on Cloud	10
Migration to Cloud Addresses Scalability in Fraud Detection for a Commercial Bank	11
Open Source Solves Integration Problems with Financial Services Systems	12
An ISV Modernizes with Open Source Software on Cloud	13
Challenges/Opportunities	14
Application Modernization with Google	16
Conclusion	19
About the Analyst	20
Message from the Sponsor	21

IDC Opinion

Digital transformation is a priority for enterprises looking to increase business agility by automating operations, delivering rich customer experiences, and launching new products and services. Today, building and sustaining a competitive advantage is linked not to size or strength but to the ability to change — to move quickly, adapt, and seize opportunities, and be agile.

This requires a mindset based on the principles of continuous innovation. Digital transformation plus continuous innovation create digital resiliency. A resilient business is not just thinking about today. It must be able to effectively address future challenges.

To be successful, organizations need to modernize the applications that their business depends on. The process replaces older technology stacks with cloud-native ones, which enable applications to be more scalable, fault-tolerant, and interoperable than their traditional counterparts. This involves implementing a microservices architecture and application programming interface (API)–driven automation. In addition, these applications need modern infrastructure that is elastic and scalable.

As organizations modernize, they adopt open source software, which can speed innovation and reduce license costs, and is supported by the community. This impacts all layers of the technology stack, including infrastructure, data management, application platforms, and development frameworks. It is no coincidence that there is a strong correlation between open source software and cloud natives. Modern cloud platforms originated from open source software, so it is logical that applications follow a similar path.

Organizations that have completed the process of application modernization by implementing open source software on cloud infrastructure report better uptime and scalability of services as well as faster time to market. Google's open source leadership and comprehensive cloud platform is helping the company's customers to be successful in their application modernization journeys.

Open Source Software on Cloud Is Accelerating Enterprise Application Modernization

Application modernization refers to updating older software to newer computing paradigms, including languages, frameworks, and agile practices to gain increased developer velocity and faster innovation cycles. Enterprises are accelerating their move to modern application development. In a 2021 IDC survey, 86% of respondents noted that their organization had modernized more than half of their legacy applications, up from 65% in 2020.

IDC data also indicates that developers have a high degree of influence and significant autonomy with respect to the selection of tools and infrastructure for projects on which they are engaged. The responsibilities of contemporary developers are no longer focused primarily on writing code but instead include the implementation of DevOps-related automation, application monitoring and life-cycle management, user experience, and security.

This is a driving factor in the shift to open source software as an alternative to proprietary technology stacks. IDC research shows that 67% of organizations are currently using open source software to support enterprise workloads. With 75% of developers reporting heavy use of Java, new development frameworks such as Quarkus, Micronaut, Helidon, and Spring Boot are accelerating the transition to cloud-native development based on containers.

The community approach to open source software increases the pace of innovation. By being open and freely available, it enables and encourages collaboration and the development of technology, solving real-world problems. It creates transparency in both access to software and the ability to act upon it. This facilitates the creation of standards and improves interoperability. It also reduces software and support subscription costs as well as vendor lock-in.

When planning for application modernization, there are three areas to consider (see Figure 1):

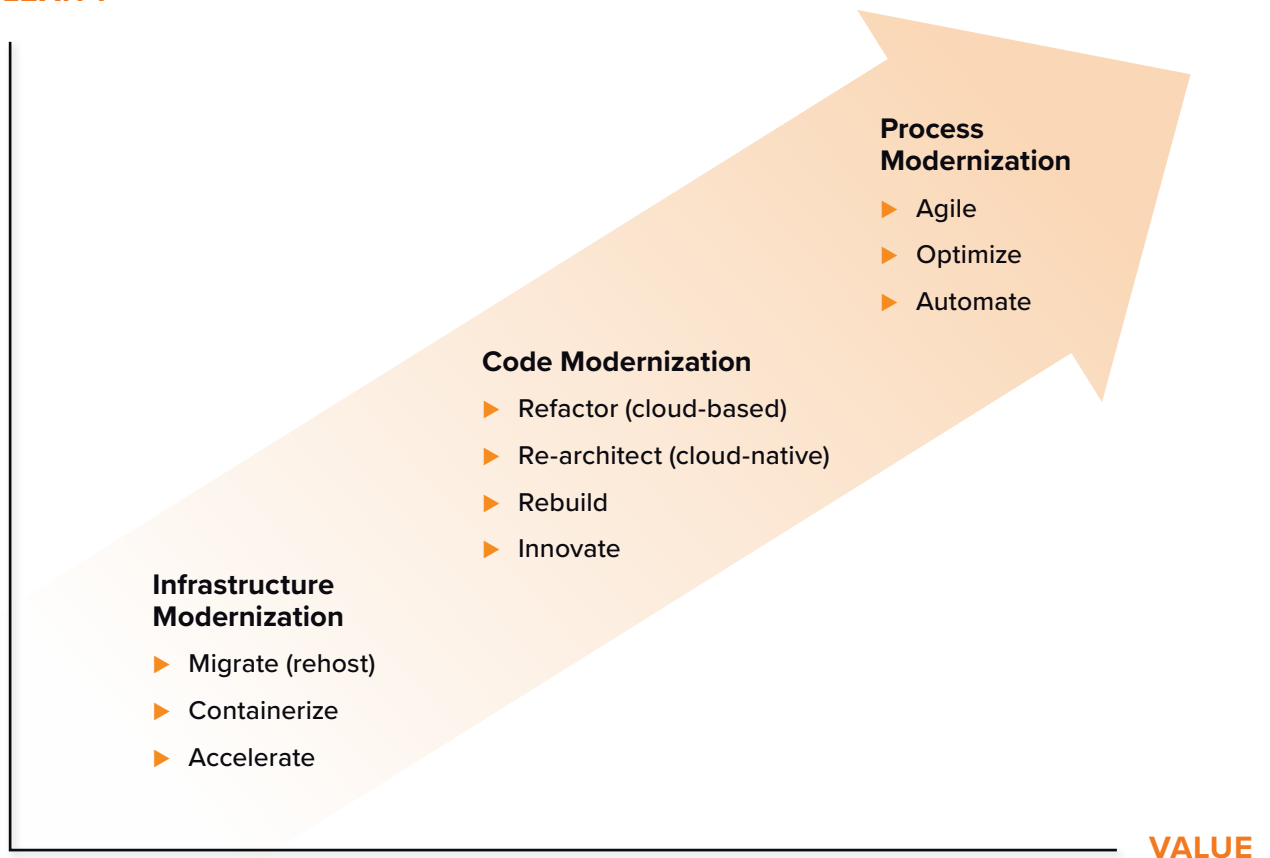
- ▶ **Infrastructure modernization** refers to implementing software-defined compute, storage, and networking resources that can scale on demand.
- ▶ **Code modernization** consists of refactoring legacy workloads using cloud-native concepts such as containers, microservices, and API-driven automation.
- ▶ **Process modernization** is the introduction of DevOps methodologies, including continuous integration and continuous delivery (CI/CD) of new software.

FIGURE 1

Three Areas of Application Modernization

(areas to consider in terms of complexity and value)

COMPLEXITY



Source: IDC, 2021

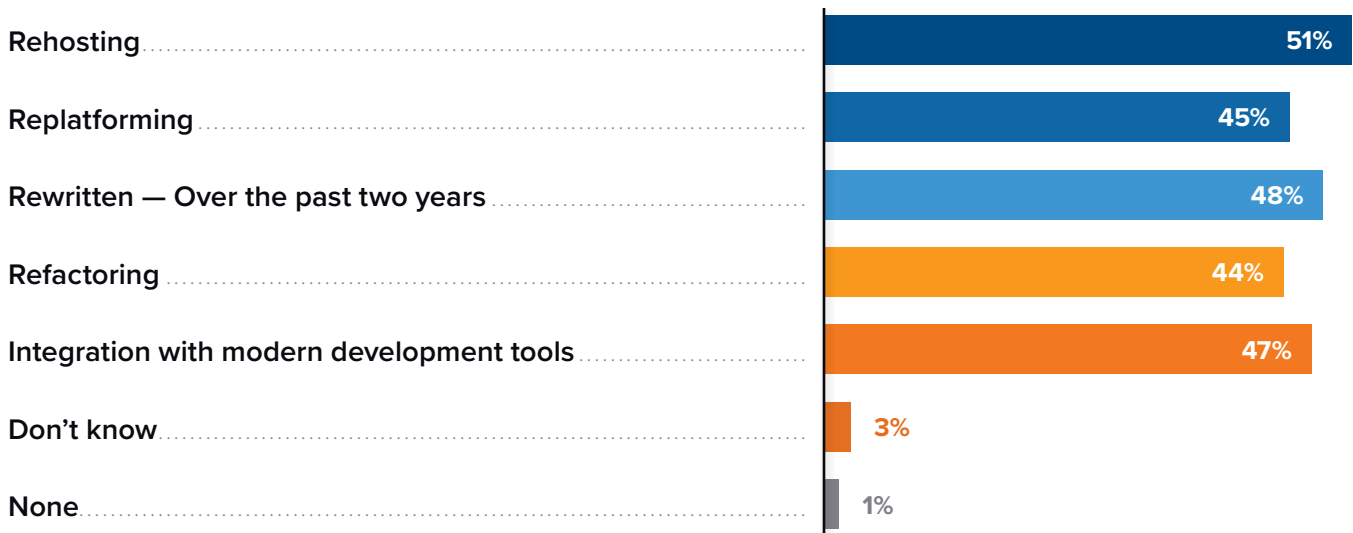
As a best practice, it is important to create an inventory of existing applications to determine both priority and the best path forward. There are five prominent modalities of application modernization:

- ▶ **Rehosting of applications:** The lift and shift of applications from one platform to the cloud, with zero to minimal architectural changes to the application
- ▶ **Replatforming applications:** The migration of applications from an on-premises environment to a cloud, with minor changes that leverage some of the cloud platform’s functionality, such as autoscaling and high availability
- ▶ **Rewriting applications:** Rewriting and coding a legacy application from scratch using modern frameworks and infrastructure
- ▶ **Refactoring applications:** The re-architecting of applications to optimize their functionality for the cloud (One common example of refactoring involves transforming monolithic applications into microservices architectures using containers.)
- ▶ **Integration with modern development tools:**
The use of modern development tools such as cloud-based integrated development environments (IDEs), DevOps toolchains, APIs, microservices, and containers to extend the functionality of legacy applications

IDC research shows that organizations are employing multiple approaches to application modernization (see **Figure 2**).

FIGURE 2
Application Modernization Approaches
(% of respondents)

Q. Which of the following modernization techniques have you used?



n = 1,932, Source: IDC, 2021

Each approach has different benefits and challenges (see **Table 1**). While rehosting is relatively simple to implement, it does not unlock the full potential of the cloud. The most benefits are achieved when developers can take advantage of cloud-native concepts such as container orchestration, serverless compute platforms, and CI/CD integration.

TABLE 1
Comparison Between Modernization Pathways

Modernization Pathway	Benefits	Disadvantages
Rehosting	Rehosting involves speed because there are minimal changes made to application architecture, and applications can be rapidly migrated to the cloud.	Applications are not optimized for the cloud and are not able to fully benefit from modern architectures, which can lead to higher operating costs.
Replatforming	Applications can more directly benefit from cloud-specific functionality such as autoscaling and other platform services.	Care must be exercised to ensure that the replatforming work does not lead to unexpected dependencies that can disrupt operations.
Rewriting	Rewriting allows developers to bring a plethora of new features and functionality to an outdated application, making it more extensible and portable.	Rewriting is time-consuming and costly and can be a complex undertaking. There is a risk that performance characteristics can vary based on the architecture.
Refactoring	Refactoring can transform the architecture of an application to render it easier to update for the long term, potentially increasing developer velocity for distributed teams.	Refactoring is complex. There are few structured guides or templates that provide developers with best practices.
Integration with modern development tools	This involves bringing modern development functionality to legacy applications with minimal disruption.	Because the foundational application remains untouched, there are limitations to the amount of new functionality that can be integrated.

Source: IDC, 2021

Technologies adopted as part of the application modernization process include the following:

- ▶ **Microservices** allow a large application to be separated into smaller independent parts, with each part having its own realm of responsibility. To serve a single-user request, a microservices-based application can call on many internal microservices to compose its response.
- ▶ **Containers** are lightweight packages of your application code together with dependencies such as specific versions of programming language runtimes and libraries required to run your software services. Modern cloud-native applications are usually built as microservices using containers.
- ▶ **Container orchestration frameworks** such as Kubernetes group containers that make up an application into logical units for easy management and discovery. It abstracts the underlying infrastructure, enabling the automation of operations and service health monitoring.
- ▶ **Functions as a service** is a serverless execution environment for building and connecting cloud services. It allows developers to write simple, single-purpose functions that are attached to events emitted from cloud infrastructure and services.
- ▶ **APIs** are programmatic interfaces that provide secure access to back-end cloud services through a well-defined representational state transfer (REST) API that is consistent across the environment, regardless of the service implementation.

Figure 3 (next page) illustrates how organizations are implementing these technologies.

FIGURE 3

Adoption of Technologies as Part of the Application Modernization Process

(% of respondents)

Q. Describe your organization’s adoption of the following technologies.

- Not using
- Considering using
- In the process of implementing
- Piloting, as a precursor to using regularly as part of our services
- Using regularly as part of our services



Source: IDC, 2021

Notes from the Field

For this study, IDC interviewed four companies that have completed the app modernization journey.

Investment Services Firm Accelerates Legacy Infrastructure Transformation by Leveraging Open Source on Cloud

IDC spoke with the IT director of an investment services firm with offices in London; Madrid, Spain; Milan, Italy; New York; and Florida. Founded in 2003, the company had amassed a range of proprietary and custom-built solutions.

While the initial implementation of these systems was successful, problems emerged over time. The company went through a phase of rapid growth, which exposed limitations in the software architecture. Systems that are critical to operations and customer interactions were experiencing performance and scalability challenges. This was compounded by pressure to provide more external access to data via web and mobile applications.

The company's software was deployed on Windows-based servers with Microsoft SQL Server in on-premises datacenters. Applications were a mix of several versions of .NET. When the primary software engineers responsible for these solutions left the company, much of the historical knowledge left as well.

Not having the skills in-house, the IT director hired a consulting firm to review the existing systems and propose a plan. Analysis showed that the existing software was not well organized and included a significant amount of technical debt. It quickly became obvious that this would not be a quick fix. The company needed a new strategy for its applications, and this was a chance to modernize on open source software in a cloud environment.

The company decided to implement MariaDB, an open source relational database designed by the original developers of MySQL. It also transitioned to PHP and a variety of open source Apache projects including Tomcat for its application layer. It chose to deploy this software on virtual machines in the cloud.

The benefits of open source software on cloud were immediate. The ability to provision resources quickly supported the rapid pace of development required to meet an aggressive timeline. It also allowed the company to try new ideas before committing to a large effort. This flexibility and agility combined with the security and scalability of the cloud allowed the company to not just address current needs but serve as a foundation for future growth.

The experience inspired the company to look beyond line-of-business applications to other areas such as productivity software. This led to the migration to Google Workspace, which includes collaboration tools such as Gmail, Calendar, and Meet along with business applications such as Docs, Sheets, and Slides. In hindsight, the IT director said that he wished he had made the move quicker: *“Don’t wait until an emergency to plan for the future.”*

Migration to Cloud Addresses Scalability in Fraud Detection for a Commercial Bank

Commercial banks are responsible for monitoring and reporting suspicious transactions in the services they provide. This is important not just to ensure regulatory compliance but also to earn customers’ trust. A program manager in charge of technology at a leading bank shared his application modernization journey with IDC.

Like most businesses today, banks look to technology as a way of differentiating themselves in the market and creating a competitive advantage. While the bank had been using cloud for many years, fraud detection systems were still deployed on local infrastructure. This was primarily to protect personally identifiable information (PII) that was deemed risky to place outside of the bank’s control.

However, these internal systems were unable to effectively handle the volume of transactions during peak times. On a monthly basis, the bank processes 70 million to 100 million transactions. During the holidays, that amount can triple. There are approximately 1,000 employees around the globe who process alerts related to these transactions.

The bank had several objectives. It needed to address its scalability problem in a secure manner. It also wanted to innovate faster in order to roll out new features and functionality. There was also a desire to reduce its investment in capital-intensive equipment and the associated software licensing and support costs.

A majority of the bank’s existing software was based on Oracle Database and Java. Instead of rehosting those applications in the cloud, the bank decided to replatform where possible and rewrite the rest so that it could take advantage of cloud-native concepts and open source software. It chose Google Cloud, implementing the Google Kubernetes Engine (GKE) and Cloud SQL for PostgreSQL. The container-based architecture allowed the bank to subdivide its larger applications into smaller units that could be developed in parallel and updated more frequently. Google provides solutions like the Oracle to Postgres toolkit to facilitate data migration using

a mix of existing open source and Google-built tooling to support the process of converting schema and establishing data replication.

Given the size of the undertaking, the bank did experience some challenges along the way, mainly related to data migration. It highlighted the importance of having a cross-functional team of technical and business analysts that understands the underlying data structures and business requirements. The bank was also moving from a waterfall to an Agile development methodology, which required training and process reinforcement.

In the end, the program manager said the effort was a success. As internal teams became more familiar with the cloud and DevOps tools, productivity accelerated. This confidence is leading to new projects with even more ambitious objectives. The bank has begun to implement Google artificial intelligence (AI) and machine learning services to make its fraud detection more accurate. According to the program manager, *“If we can improve our detection algorithms by just 0.05%, it has a meaningful impact to our business.”*

Open Source Solves Integration Problems with Financial Services Systems

Financial services organizations deal with large volumes of data. While that is a challenge in itself, imagine the complexity of that data being spread over hundreds of systems. That is what was facing a financial services organization’s vice president (VP) of technology, who decided to utilize the cloud as an alternative to in-house systems to ensure its data integrity.

This particular use case involved credit limits associated with business loans. When borrowers apply for a loan, a series of checks are required to validate credit-worthiness and to ensure borrowers do not exceed their approved credit line. Although this appears straightforward, there are multiple systems involved that are not always in sync with one another.

For example, loan approval occurs in one system, loan booking in another, and loan servicing in yet another. There are 30 individuals across the United States, the United Kingdom, Canada, and Mexico who are responsible for validating this data, and what began as a small problem was starting to grow. In some situations, borrowers were being over-approved for a loan, while others would experience underwriting delays. Reconciliations were mainly a manual process that was prone to user error.

The company had not only a large investment in IBM mainframes but also a variety of other data stores, including Microsoft, Oracle, and Sybase. Applications ran in IBM WebSphere, Oracle WebLogic, and Microsoft IIS. Different integration strategies were used to synchronize data between systems, but there was a lack of consistency in how logic was applied, and it was becoming increasingly difficult to manage and fix.

The VP investigated cloud service providers for a solution. He was looking for a better way of aggregating this information and wanted to employ modern tools for data management, analysis, and integration. After evaluating several cloud SPs, Google Cloud was selected. The primary reasons given for selecting Google Cloud were the maturity of the platform, Google’s experience delivering on the needs of large companies, and a focus on the financial services industry.

Google's leadership in the open source community and support for open source software was another factor. Existing proprietary databases could be migrated to Google Cloud SQL, WebSphere and WebLogic applications migrated to Cloud Run and GKE, and .NET applications running in IIS migrated to .NET Core and Kestrel. Moving to open source software on Google Cloud provided an opportunity to standardize on a new software stack that would serve as a foundation for future growth.

The team also implemented several Google technologies including BigQuery and Data Studio. It found that developing for the cloud consumed less time than traditional environments, and the elasticity enabled the company to scale the environment as needed. The data visualization tools eventually replaced other licensed tools.

When asked about what could have gone better, the VP said that a majority of the challenges were related to business processes and culture rather than technical. For example, while connecting an on-premises data source to the cloud is relatively straightforward, there were internal hurdles in obtaining the necessary security approvals. There was also training required to transition mainframe developers to the cloud environment.

The company expects to accelerate its use of cloud going forward. Now that the groundwork has been laid, new projects can be initiated and executed quickly. *"Before the cloud, it would take six months to get a new server ordered and provisioned,"* said the VP. *"Now, it only takes minutes."*

An ISV Modernizes with Open Source Software on Cloud

Enterprises are not the only ones dealing with a transition from legacy applications to modern applications. Independent software vendors (ISVs) must also update their products to meet growing customer requirements. IDC met with the chief technology officer (CTO) of a European ISV based in Norway to understand this perspective.

This company offered a customer relationship management (CRM) product with 10,000 users. It was originally written in C++ and COM as a 32-bit application. The software was deployed in customer facilities on Microsoft infrastructure and required a local install for each Windows client. To address scalability and performance issues, there was an attempt to use a Citrix-based virtualized environment, but it did not resolve the issues.

The CTO had recently joined the organization to remediate the problems with the application and identified several concerns. In addition to performance, there were reliability issues. Since the software was deployed on traditional infrastructure, it often meant sending engineers onsite to troubleshoot. To make matters worse, many of the libraries used by the application were no longer supported, and it was hard finding developers who wanted to work on such an old system.

The decision was made to rewrite the application using open source software on the cloud. The goal was to use as much serverless technology as possible, augmenting with containers where needed. This involved introducing object storage, NoSQL databases, and infrastructure as code. The company implemented PostgreSQL, MongoDB, and Redis for data management. Applications were written in JavaScript and Python.

By moving to open source software, the company was able to eliminate much of its proprietary software costs. It also enabled the company to audit code and customize where needed. The company avoided “recreating the wheel” and focused time on value-added functionality.

The new product was now multitenant, which was important as the company optimized cloud costs. Customers received an improved experience with better performance and availability. The ISV benefited from having better observability of the environment, including how the application was being used. The usability data provided insights into product road maps to proactively identify new features and fixes.

The biggest challenge was managing change both internally and with customers. Like others in this study, new skills were needed, and there was some trial and error in implementing new cloud services. Customers had some initial apprehension moving their data to the cloud, but the positives far outnumbered any concerns.

The CTO has no regrets. He was able to help the company modernize its key product while reducing costs and improving customer satisfaction. When asked for advice for others just starting an application modernization journey, he said, *“There is more than one way to achieve your goals in the cloud. Leverage experienced cloud architects to help you make the right choices.”*

Challenges/Opportunities

Despite the increased momentum of organizations’ executing on application modernization strategies, it is not without challenges. However, the biggest obstacle is not the technology itself. It requires a cultural shift that involves a new way of thinking about application architecture and infrastructure design.

Application modernization initiatives are rarely successful if done in an ad hoc manner. It is critical to obtain management support to overcome common challenges such as budget, priority, and scheduling. Like any major project, there must be clear objectives that can be measured and tracked.

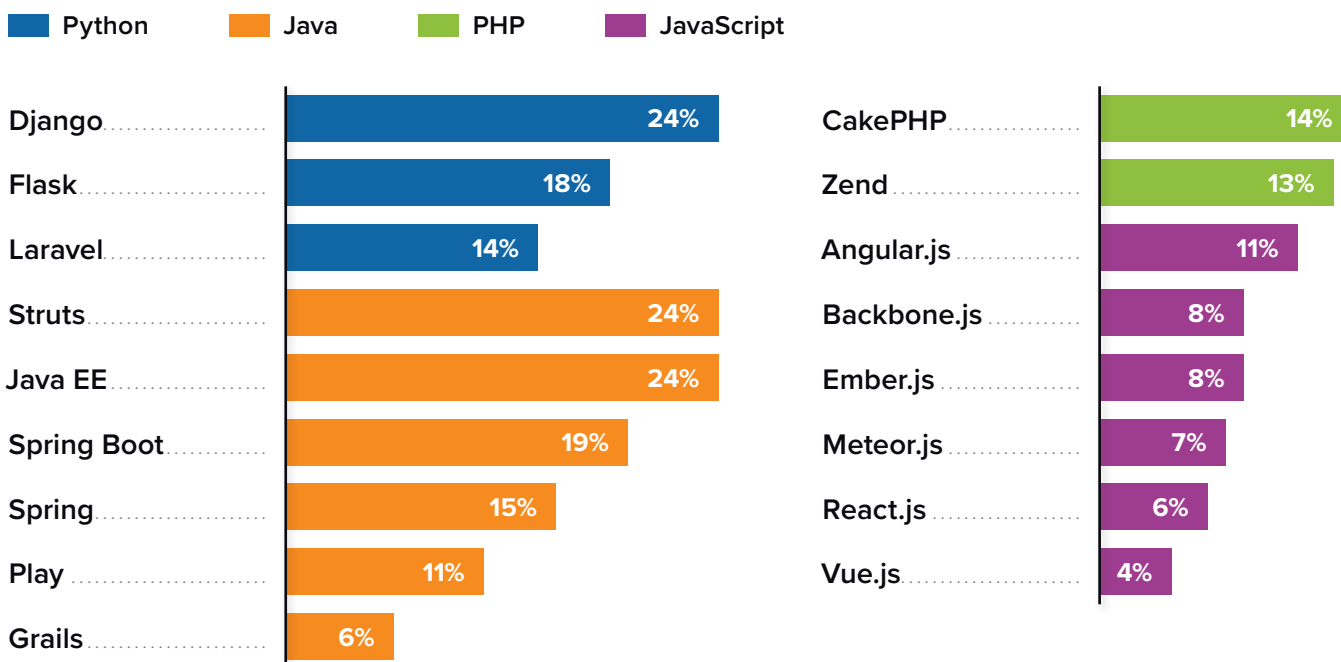
While skill gaps are often cited as a primary reason for not adopting cloud-native concepts, the proliferation of open source software has created a wealth of new cloud-native developers who are familiar with modern languages and frameworks. **Figure 4** (next page) shows the most popular Python, Java, PHP, and JavaScript frameworks being used by organizations, which are led by Django, Struts, CakePHP, and Angular.js, respectively.

FIGURE 4

Development Frameworks Used by Organizations

(% of full-time developers)

Q. Which of the following development frameworks have you or your team used over the past 12 months for at least 20% of applications you developed during this time period?



Source: IDC, 2021

This new breed of developers is also the driving force in the implementation of DevOps tools that automate processes in the development lifecycle, inclusive of those that streamline collaboration between development and operations teams. This means that automation is increasingly part of development-related operations, and as a result, developers can deploy code and update applications more frequently than ever before. IDC data shows that 98.6% of developers are working on projects that either have adopted or are planning to leverage DevOps practices.

Cloud service providers offer many resources that can help with cloud adoption using open source software. This includes online documentation with reference architectures and code samples. By following existing development and deployment patterns, developers can quickly become productive.

Cloud SPs also have training materials and workshops to accelerate learning. In some cases, there are programs where cloud solutions architects review existing systems and offer recommendations on how best to modernize. This is often accompanied with service credits or other incentives to help with workload migrations.

Partners also play a role. These are companies that consist of cloud-certified resources that can assist with application modernization projects through professional services engagements or preconfigured solutions.

Application Modernization with Google

Google Cloud (GC) offers a wide array of cloud computing services, running on the very same infrastructure that Google uses to power end-user products such as Google Maps, YouTube, and Google Search. This includes services like compute, storage, networking, data analytics, and AI, which, along with embedded management tools, can be used to develop, test, and deploy new applications and features quickly.

Google is empowering developers to build, run, and operate cloud-native applications by embracing modern approaches like serverless, microservices, and containers. GC enables developers to quickly code, build, deploy, and manage without compromising security or quality. It also facilitates the deployment of modern applications in hybrid cloud and multicloud environments with Google Anthos — a modern application management platform that provides a consistent development and operations experience for cloud and on-premises environments.

In *The Business Value of Improved Performance and Efficiency with Google Cloud Platform* (IDC #US46610620, June 2020), IDC performed in-depth interviews with organizations using GC to understand its impact on their businesses, IT operations, and costs. Interviewed customers described achieving strong value with GC by obtaining the performance, reliability, and scalability their businesses require on a cloud platform that is cost-effective and efficient to operate.

Based on these interviews, IDC calculated that these Google customers will achieve benefits worth \$103,200 per \$1 million in annual revenue by:

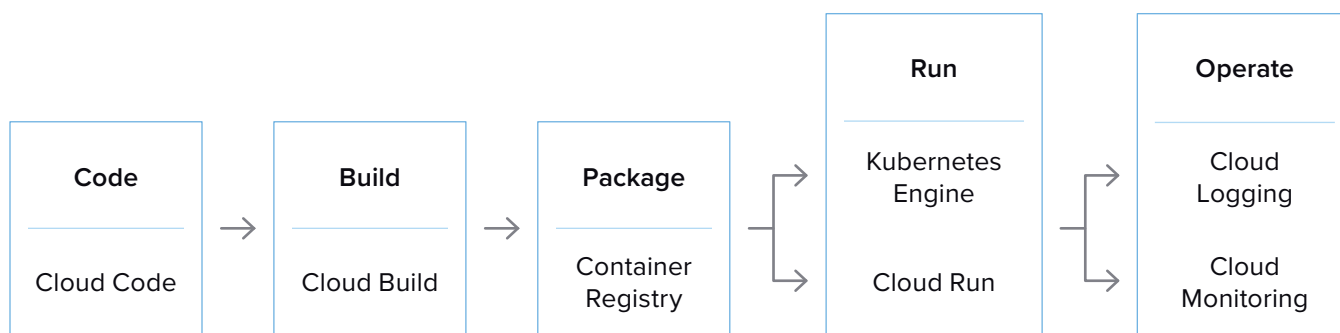
- ▶ **Empowering application development teams** through improved infrastructure agility, scalability, flexible capacity, and other built-in platform functionality
- ▶ **Enabling business growth** by having an IT infrastructure that allows for addressing and expanding revenue opportunities
- ▶ **Maximizing the value of IT staff time** by leveraging automation, embedded technologies, and Google support
- ▶ **Optimizing IT infrastructure costs** through platform functionalities such as autoscaling and technologies such as the Google Kubernetes Engine

Other business value highlights include:

- ▶ 36% lower three-year cost of operations
- ▶ 16% higher revenue per organization per year
- ▶ 41% more efficient IT teams
- ▶ 19% higher developer productivity
- ▶ 26% lower IT infrastructure costs

GC accelerates engineering productivity and simplifies operations with end-to-end tooling (see **Figure 5**). Automatically build, test, deploy, and manage code changes across Google Kubernetes Engine and Cloud Run for a serverless runtime.

FIGURE 5
Google Cloud’s End-to-End Tooling for Cloud-Native App Development



Source: Google, 2021

Security and compliance across all artifacts are implemented with security embedded in the software delivery process that identifies vulnerabilities as soon as code is committed. It is possible to set up automated real-time deployment policy checks to gain tight control and ensure that only trusted artifacts get deployed.

GC’s highly responsive infrastructure with built-in best practices includes serverless tooling that automatically scales up and down depending on traffic. Built-in CI/CD integrations and deployment practices like traffic splitting reduce the risk of deploying a new version of software into production.

The platform also easily connects to popular tools, including GitLab, Jenkins, Terraform, Datadog, and Prometheus. These integrations enable a transition to GC at your own pace.

A sampling of Google's cloud-native app development services:

- ▶ **Cloud Code** provides IDE support for the full development cycle of Kubernetes and Cloud Run applications, from creating and customizing a new application from sample templates to running a finished application.
- ▶ **Cloud Build** is a service that executes builds on Google Cloud infrastructure. Cloud Build can import source code from Cloud Storage, Cloud Source Repositories, GitHub, or Bitbucket; execute a build to your specifications; and produce artifacts such as Docker containers or Java archives.
- ▶ **Artifact Registry** is a fully managed service with support for both container images and non-container artifacts such as Helm charts. It can also store and manage Java, Node.js, Python, Debian, and RPM Linux packages.
- ▶ **Google Kubernetes Engine** provides a managed environment for deploying, managing, and scaling containerized applications using Google infrastructure. The GKE environment consists of multiple machines (specifically Compute Engine instances) grouped together to form a cluster.
- ▶ **Cloud Run** is a managed compute platform that enables organizations to run containers that are invocable via requests or events. Cloud Run is serverless: it abstracts away all infrastructure management.
- ▶ **Operations Suite** is a fully managed service that collects metrics, logs, and traces across Google Cloud; includes built-in dashboards and views for monitoring the platform and applications; and performs alerts and notifications.

Conclusion

A modern organization must be digitally resilient to quickly respond to changing customer requirements, competitive threats, or operational challenges. This requires infrastructure that can scale on demand, applications based on a microservices architecture, and agile software development practices.

A cloud-native mindset is key to application modernization, enabling organizations to innovate faster while improving overall performance and reliability. While there are different approaches to migrating an application to the cloud, each approach is valid depending on requirements and the desired outcome.

This also creates an opportunity to explore the benefits of open source software as an alternative to proprietary frameworks. In addition to reduced license costs, organizations appreciate the pace of innovation, transparency, and community-based support that open source provides.

The trends are clear. A majority of organizations are investing heavily in application modernization initiatives leveraging open source software on cloud infrastructure. The strategic nature of these efforts has elevated the role of the developer in the eyes of senior management as essential in achieving business goals. Google has established itself as a leader in application modernization through its enterprise-grade cloud infrastructure, open source–friendly platform services, and developer focus, making it the preferred choice for both start-ups and multinational corporations.

About the Analyst



Dave McCarthy
Research Vice President,
Cloud and Edge Infrastructure Services, IDC

Dave McCarthy is a Vice President within IDC's worldwide infrastructure practice, where he leads a team of analysts covering shared (public) cloud, dedicated (private) cloud, and edge strategies.

Benefitting both technology suppliers and IT decision makers, Dave's insights delve into how hybrid and distributed cloud platforms provide the foundation for next-generation workloads, enabling organizations to innovate faster, automate operations, and achieve digital resiliency.

His research is available via syndicated research programs (subscription services), data products (IDC Trackers) and custom engagements.

[More about Dave McCarthy](#)

Message from the Sponsor

About Google Cloud

Google Cloud provides organizations with leading infrastructure, platform capabilities and industry solutions. We deliver enterprise-grade cloud solutions that leverage Google's cutting edge technology to help companies operate more efficiently and adapt to changing needs, giving customers a foundation for the future. Customers in more than 150 countries turn to Google Cloud as their trusted partner to solve their most critical business problems.

[Learn more](#)

IDC Custom Solutions

This publication was produced by IDC Custom Solutions. As a premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications, and consumer technology markets, IDC's Custom Solutions group helps clients plan, market, sell and succeed in the global marketplace. We create actionable market intelligence and influential content marketing programs that yield measurable results.



 @idc

 @idc

[idc.com](https://www.idc.com)

© 2022 IDC Research, Inc. IDC materials are licensed [for external use](#), and in no way does the use or publication of IDC research indicate IDC's endorsement of the sponsor's or licensee's products or strategies.

[Privacy Policy](#) | [CCPA](#)