Accelerating Developer Productivity and Agility with Google

Arnal Dayaratna | *September 2018*

An IDC White Paper, Sponsored by: Google Cloud

IDC OPINION

Overview of the Contemporary Development Landscape

Business needs for digital solutions have intensified the urgency for rapid application development and deployment. Today's software development landscape is characterized by a broad and varied array of development languages, tools, methodologies, and frameworks. Developers must be proficient in many languages, frameworks, and development environments. Moreover, widespread adoption of cloud computing has revolutionized the IT environment used by both enterprises and consumers. The IT environment requires developers to illustrate familiarity with deployment platforms such as public clouds, private clouds, hybrid cloud topologies, containers, microservices, and function as a service (FaaS).

The acceleration of digital transformation has increased the availability of developer tools and deployment environments. This expansion of independent platforms, frameworks, and infrastructures used for software development has led to a corresponding increase in the complexity of navigating the ecosystem of processes and technologies available. The profusion of development languages, frameworks, integrated development environments, and deployment infrastructures has made contemporary application development extremely complex. Moreover, growth of open source software has required developers to reflect on the methods of monetizing open source technologies, integrating open source into their applications, and using open source to innovate and drive business growth.

On the other hand, the vibrancy of the DevOps movement and the multiplicity of development tools and infrastructures mean that developers enjoy greater decision-making authority in the development of digital solutions. Part of this decision-making authority extends to the selection of appropriate tools and frameworks for application development and management.

Developers not only confront the challenge of shipping software faster but also encounter competing challenges to ensure software quality and the appropriate mitigation of risks associated with rapid development and deployment. For example, even though shipping faster is the key to survival, developers cannot afford to risk compromising the security of their applications.

This white paper argues that FaaS, continuous integration/continuous delivery (CI/CD), and application life-cycle management (ALM) technologies accelerate developer productivity and agility by providing developers with a microservices-style, fully managed infrastructure for application development. FaaS applications take advantage of the elasticity, scalability, and pay-per-use attributes of the public cloud. Meanwhile, the implementation of continuous integration/continuous delivery automates upgrades to software applications and enhances the implementation of quality assurance processes in the software development life cycle. Application life-cycle management tools further assist developers in proactively identifying and remediating issues.

IDC believes that the use of FaaS in conjunction with CI/CD and application life-cycle management technologies accelerates application development by enriching, streamlining, and simplifying the development experience. When used with a scalable infrastructure such as the public cloud, FaaS, CI/CD, and ALM technologies let developers focus on application design and development, alleviating the operational challenges of managing application updates, infrastructure scalability, and the implementation of manual processes for quality assurance and quality control.

IDC further notes that FaaS applications, CI/CD solutions, and application life-cycle management technologies enhance frameworks that improve the ability of developers to scale their applications as needed. FaaS, for example, supports the horizontal scaling of applications, whereas the automation delivered by CI/CD and ALM tools empowers developers to build applications that can transition from zero user to millions of users without re-architecting the application. CI/CD improves the ability of applications to scale by automating the testing of application scalability. Meanwhile, ALM tools empower developers to scale applications by monitoring metrics related to application and infrastructure performance as an application scales up or down.

SITUATION OVERVIEW

The development space is the proliferation of technologies that enhance software development by adding value to existing applications. For example, contemporary developers need to become proficient in using technologies and tools related to embedded analytics, data visualization, IoT, chatbots, voice-activated software, APIs, and tools for data management and big data. In addition, the growing adoption of artificial intelligence, machine learning, and deep learning technologies has required developers of all types of applications to become familiar with these technologies.

Furthermore, deployment infrastructures such as platform as a service (PaaS) and containers expand developers' options, with preconfigured frameworks for conversational apps, image recognition software, text-to-speech software, predictive analytics, and IoT applications.

Developers that once had to build applications entirely from scratch now enjoy an embarrassment of riches, with advanced capabilities and prebuilt features immediately available for rapid development and deployment. But velocity loses its value when quality suffers. To move quickly and with confidence, developers must use tools that automate the full software development life cycle and work across platforms and operating systems.

FUTURE OUTLOOK

Scenarios

FaaS is a cloud platform that allows developers to develop, deploy, and manage event-driven applications without the responsibility of managing the infrastructure on which the applications run. Because FaaS executes in response to events, FaaS applications use compute infrastructure resources only for the duration required to execute code triggered by an event.

By using infrastructure only in response to event-driven triggers, FaaS deployments take advantage of the economics of pay-per-use cloud computing to reduce the cost of application hosting. While non-FaaS applications are hosted in the cloud by means of a persistent hosting model, FaaS enables customers to pay only for the execution of functions that handle discrete events.

Separate from their economic benefits to customers, FaaS empowers customers to use a modular approach, in which discrete functions take responsibility for specific components of the application. This modular architecture resembles that of microservices-based applications. It provides FaaS applications with the benefits associated with microservices, such as accelerated upgrades, testing, bug fixes, and root cause analysis for application life-cycle management purposes.

The modularity of FaaS also enhances developer productivity by enabling the rapid development of discrete functions, thus facilitating the execution of parallel development streams so that developers can work on many different functions in parallel. These functions can also be reused across different applications so that developers can rapidly build on the functionality others have created.

While FaaS enhances developer productivity and agility, the infrastructure that enables the implementation of FaaS remains inchoate. For example, FaaS platforms are evolving to provide developers with greater visibility into dependencies and relationships between functions. In addition, FaaS platforms would benefit from enhanced functionality for functions to reference functions.

FaaS can improve application life-cycle management and increase visibility into how effectively applications are executed. This, in turn, helps developers pinpoint a specific function that is causing performance issues.

Scalability

While FaaS provides ample potential to transform and revolutionize application development, scalability is another key consideration. Scalability automates the ability of applications to dynamically and intelligently modify their consumption of infrastructure resources in conjunction with the demands of end users.

While scalability is one of the key attributes of cloud computing and cloud-based applications, it is also a key attribute of container-based applications that leverage container orchestration frameworks such as Kubernetes. Kubernetes automates the deployment, scaling, and operational management of container-based applications, thereby allowing developers to focus on using containers to build container-based applications. This capability of Kubernetes to automate the scaling of container-based applications means that developers need not concern themselves with re-platforming applications as they experience more or less usage.

Scalability can be implemented horizontally and vertically. Horizontal scaling refers to the addition or subtraction of machines into the pool of available resources. Vertical scaling refers to changes in the computing power specific to a static universe of computing resources. FaaS applications enjoy the benefits of automated horizontal scaling; the scalability of these applications is inherently managed by the FaaS platform on which they operate.

Continuous Integration/Continuous Delivery

CI/CD refers to technology and processes that perform two functions: CI focuses on integrating code contributions from individual developers into a unified codeset, and CD refers to making the release process of updates to software, such as patches, upgrades, and bug fixes, safe, low risk, and quick.

CI facilitates the timely detection of bugs and addresses integration challenges prior to deployment. CI processes typically execute several times each day with the purpose of detecting problems early and often. Meanwhile, CD automates deployments across multiple targets and environments with a focus on making the releases more reliable and less time intensive. One of the other key advantages of implementing CD is the ability to reliably roll back an application to an earlier state if a defect or an anomalous behavior is identified.

When implemented properly, CI/CD processes accelerate the pace of software releases and automate the detection and remediation of defective software components to reduce risk. Because of their ability to improve quality, accelerate software releases, and foster collaboration, the CI/CD processes enhance developer productivity and agility in keeping with today's requirement for rapid development and deployment.

Key drivers for the implementation of CI/CD include:

- **Developer productivity.** CI helps developers proactively catch issues through automated build and test so that they can focus on more complex software development problems rather than debugging and resolving integration problems. CI enables developers to spend more time building features and less time finding issues.
- **Release of new features quickly and at low risk.** Customers have increasingly come to expect frequent updates and enhancements to their applications. By leveraging best-in-class deployment best practices, such as immutable infrastructure and canary deployments, CD reduces the risk of releasing these quick customer updates into production.
- **Competitive advantage.** Another reason organizations would do well to implement CI/CD processes is their chance to surpass the competition by responding to market needs faster. As organizations increasingly integrate CI/CD processes into their software delivery process, they are able to incorporate updates to their applications several times a day.
- **Integration of processes for implementing security into the development life cycle.** An additional benefit of CI/CD processes is the integration of security concerns into the development process. When properly implemented, CI/CD requires organizations to reflect on security-related considerations at every stage of the development process, given that CI/CD processes evaluate code submissions from many individuals, teams, and stakeholders.
- **Automation of scalability testing.** CI/CD frameworks empower developers to integrate scalability testing across different application components. The addition of scalability testing to CI/CD protocols enables developers to perform continuous scalability testing that tracks the ability of an application to scale to accommodate the pressures imposed on the application by end users.

Google has innovated the use of CI/CD tools by providing developer tools that help organizations release software at high velocity without sacrificing security or quality. Specifically, Google Cloud Platform (GCP) developer tools help set up end-to-end automation from source to production across multicloud, hybrid, and on-premise environments.

Seamless Application Management

While CI/CD improves developer productivity and agility in the production and deployment of software, application life-cycle management empowers developers to more effectively monitor and manage the health of applications. Defined as the management of the life cycle of an application from its conception through the phases of development, deployment, and testing, application life-cycle management standardizes communication and collaboration protocols between software development team members and automates processes related to software development and deployment.

Application life-cycle management spans disciplines that include program and portfolio management, project management, requirements definition and management, software development, software testing and quality assurance, and post-deployment monitoring and management. Application monitoring and management have traditionally focused on evaluating application performance against key health metrics.

Application monitoring tools traditionally feature dashboards, alerts, and notifications that deliver real-time visibility into the health of an application. In recent years, application monitoring has additionally included infrastructure monitoring and, notably, the connection of infrastructure and application monitoring.

In addition, application life-cycle management tools improve scalability by automating application performance management related to CPU consumption, memory consumption, latency, and other parameters. ALM tools give developers the ability to scale either the whole or parts of their application. Google's Stackdriver provides monitoring and diagnostics for applications hosted on GCP and Amazon Web Services (AWS). Stackdriver delivers monitoring, log aggregation, error reporting, debugging, and tracing to cloud-based application deployments. Stackdriver's suite of functionality for tracking metrics, tracing functionality, and providing log management is integrated into dashboards. Data-driven alerts accelerate the time to insight for developers looking to diagnose and solve performance issues. In addition, Stackdriver provides a service called Stackdriver Profiler that examines the impact of a line of code or a function on overall application performance.

Importantly, Stackdriver Profiler can assess the impact of functions and lines of code on applications that are deployed in production with minimal interference to the operation of the application. Stackdriver Profiler's ability to understand the impact of code across multiple deployment infrastructures enables the tool to assess how code performs across different environments.

Stackdriver's broader ability to support application monitoring across more than one public cloud vendor represents one of the key differentiators of the tool in the application monitoring space. Customers can use Stackdriver to monitor multicloud deployments across a multitude of cloud vendors while obtaining an integrated picture of application performance via an interactive dashboard with drilldown functionality.

Stackdriver increases developer productivity by providing insight into application performance through a collection of tools and analytic methodologies. Customers can use Stackdriver to seamlessly manage applications, monitor performance, and swiftly go deeper to examine performance issues or to implement remediation efforts as needed.

KEY GOOGLE ACCELERATORS OF DEVELOPER PRODUCTIVITY AND AGILITY

FaaS solutions are still relatively new and evolving. Google Cloud Functions, for example, has yet to achieve integration with the vast majority of Google's products and services. But Google's FaaS technology stands poised to stand out from competitors because of its ability to harness Google's strengths in artificial intelligence, machine learning, data management, and data analytics. Google also supports serverless containers, which allows Docker images to be used as inputs for FaaS applications. In addition, the depth of Google's investments in Kubernetes technologies promises to empower customers to develop FaaS that leverage container-based architectures as opposed to virtual machines (VMs) for their foundational infrastructure.

Google's leadership in big data, data analytics, and machine learning positions customers of the company's FaaS solution to develop cloud functions that are differentiated based on their sophisticated capabilities. For example, users of Google Cloud Functions can harness the platform to build applications that leverage Google's solutions for image recognition and natural language processing.

The openness of Google's platform to third-party CI/CD technologies promises to give developers flexibility in integrating CI/CD processes with FaaS. For example, CircleCI integrates with GitHub and deploys on GCP for FaaS as well as other use cases. GCP's elasticity, scalability, and pay-per-use functionality free up developers from having to manage infrastructure while empowering them to focus more time and energy on application development.

Benefits of using GCP developer tools for CI/CD include:

- Reduction of manual processes throughout the development and deployment cycles as a result of automated build, test, artifact management, and deployment tools
- Automated rollbacks as a result of advanced release strategies such as traffic splitting, blue-green deployments, and automated canary analysis to improve the safety of application rollouts
- Flexibility to deploy GCP on-premise or on other cloud platforms
- Enhancement of security by identifying package vulnerabilities in container images; collect, store, query, and retrieve artifact metadata or set up manual approvals and real-time deployment policies

Figure 1 illustrates a reference CI/CD pipeline using GCP developer tools along with Spinnaker and Grafeas.

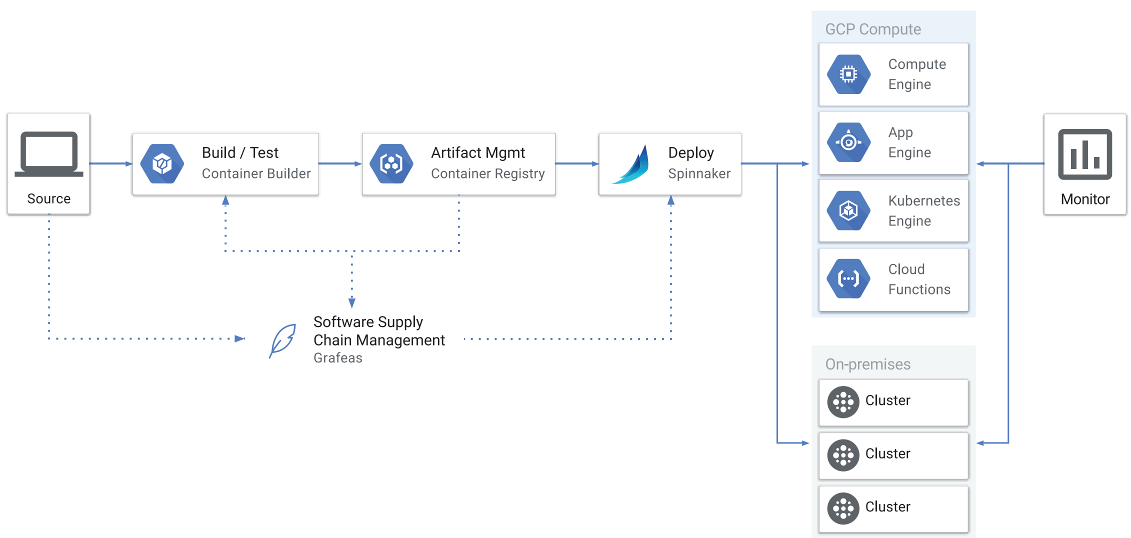
For example, customers can use Google Cloud Platform developer tools to implement a CI/CD workflow on Google Kubernetes Engine (GKE) clusters, Google Compute Engine container-optimized VMs, or other hybrid and multicloud environments.

GCP tools also let developers use an end-to-end CI/CD pipeline across multicloud, hybrid cloud, or on-premise environments. Spinnaker, an open source, multicloud CD platform, can be used with Google Container Builder and Container Registry to set up a fast, safe, repeatable deployment process. Grafeas, an open artifact metadata API, helps store, query, and retrieve artifact metadata across the software development life cycle. Grafeas lets teams set up policy controls to prevent the deployment of vulnerable images on on-premise, private, and public cloud clusters.

Developers can choose from numerous integrations across the source, build, test, deploy, and monitoring stages to set up a CI/CD pipeline. Examples include GitHub, Bitbucket, Jenkins, Travis CI, CircleCI, Docker Hub, Artifactory, Chef, Terraform, Puppet, Datadog, and Prometheus.

Figure 1

Reference CI/CD Pipeline Using GCP Developer Tools



Source: Google, 2018

CHALLENGES/OPPORTUNITIES

Challenges and opportunities for FaaS technologies involve managing the complexity of designing applications that use a multitude of functions. As the number of functions used by an application increases, developers will benefit from access to frameworks that enable them to manage and configure interrelationships between those functions. Developer tools that provide insight into relationships and dependencies between functions within an application not only accelerate the development of FaaS applications but can also improve the quality and scalability of FaaS applications.

For example, developer tools that provide developers with an understanding of how one change to a function in a multifunction application affects the entire application will provide valuable insights. Other opportunities include the enhancement of application life-cycle management functionality for understanding how the performance of a FaaS application is dependent on one or more functions.

CONCLUSION

In recent years, the profusion of contemporary developer tools, development languages and frameworks, and deployment infrastructures has made the software development landscape increasingly complex. The wide variety of different development platforms from different vendors with often different licensing models have led to increased fragmentation within the universe of available developer tools. Open source development technologies have eased access to developer tools and further expanded the options and paths available to developers.

Seismic changes in the geographic distribution of developers — with the increasing concentration of developers in the Asia/Pacific region (especially in India and China) — make tools that simplify the development experience and facilitate collaboration between global team members very attractive.

FaaS can help lessen complexity by empowering developers to build applications out of discrete functions and without worrying about the underlying infrastructure. Google's offerings, with the company's automated scalability, CI/CD, and application management features, promise to further augment the capabilities of Google Cloud Functions, Google's proprietary FaaS development platform.

Google's investments in Kubernetes provide a robust framework for integrating scalability, CI/CD, and application monitoring and management into the development of FaaS by taking advantage of the capabilities of the Google Kubernetes Engine to integrate with CI/CD tools to automate builds and releases while drawing upon the scaling and orchestration functionality of Kubernetes. Google's leadership in Kubernetes promises to differentiate the company's FaaS offering through augmented CI/CD and application management functionalities that improve developer productivity and agility by simplifying the FaaS development experience and the broader application development experience.

ANALYZE THE FUTURE

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street

Framingham, MA 01701

USA

508.872.8200

Twitter: @IDC

idc-community.com

www.idc.com

Copyright Notice