A vibrant illustration of a green circuit board with various components like capacitors and resistors. Numerous small, pink and blue ants are crawling across the board, following paths that resemble circuit traces. The overall aesthetic is futuristic and tech-oriented.

# THE STATE OF SERVERLESS

## UPDATED JUNE 2022

This research builds on the [previous edition](#) of this article, which was published in May 2021.

Serverless has transformed application development by eliminating the need to provision and manage any underlying infrastructure. The current serverless ecosystem has grown more mature, and it now has considerable overlap with the world of container-based technologies. The wide range of available options has led over half of organizations operating in each cloud to adopt serverless.

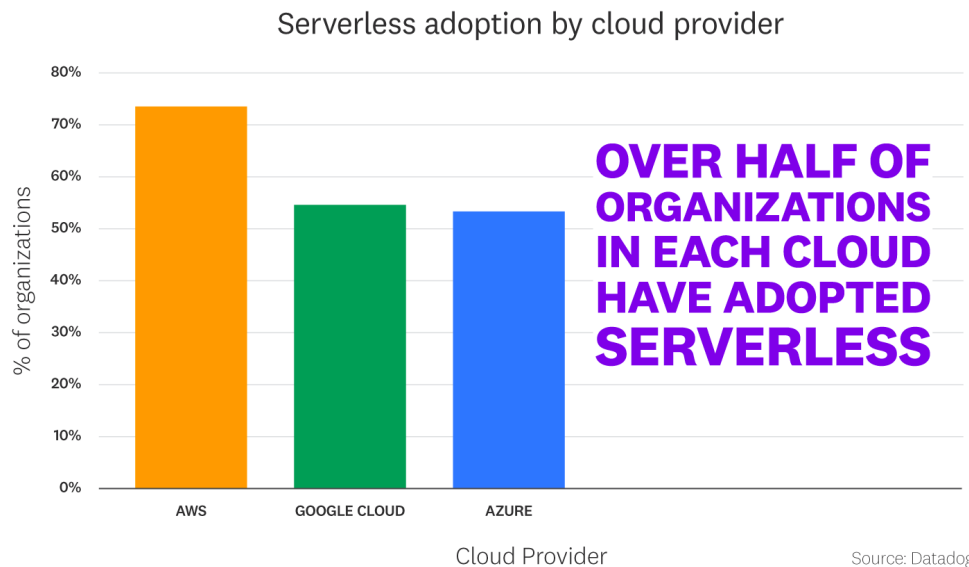
For this report, we examined telemetry data from thousands of companies' serverless applications, and we identified three key themes in how teams are using serverless today. First, serverless compute has become an essential part of the technology stacks of organizations that operate in each cloud. Second, AWS Lambda remains extremely popular among AWS customers, who are using it in new ways to support their unique business needs. Finally, there are meaningful differences between the serverless offerings available within AWS, Azure, and Google Cloud, and each gives users distinct options for building serverless applications.

Continue reading to further explore these trends within the serverless landscape.



## Over half of organizations operating in each cloud have adopted serverless

In 2020, we reported that half of Datadog's customers using AWS had adopted Lambda to run event-driven code with minimal operational overhead. As the popularity of FaaS products like AWS Lambda continues to grow, we have also seen a significant increase in adoption of other types of serverless technologies offered by Azure, Google Cloud, and AWS.



This evolution highlights the growing range of options available to organizations who want to go serverless, as well as a shift in how serverless technologies are leveraged. For instance, in addition to using individual serverless functions to run event-driven code, many organizations are also deploying containerized applications on serverless platforms such as Azure Container Instances, Google Cloud Run, and Amazon ECS Fargate. These services offer a range of benefits, which we will explore in more detail later in this report.

Note: For the purpose of this fact, a serverless organization uses at least one of the following technologies:

**AWS:** AWS Lambda, AWS App Runner, ECS Fargate, EKS Fargate

**Azure:** Azure Functions, AKS running on Azure Container Instances

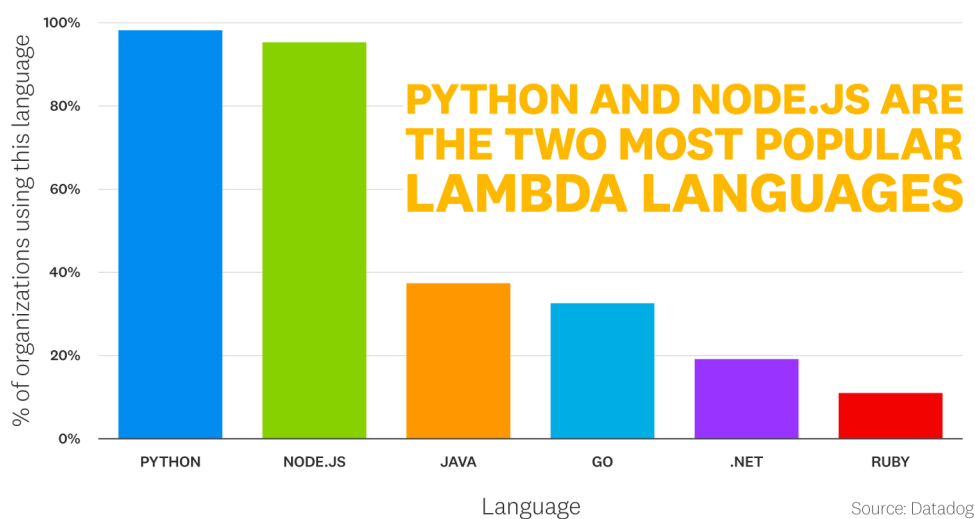
**Google Cloud:** Google Cloud Functions, Google App Engine, Google Cloud Run



## Python and Node.js remain dominant among Lambda users

Python and Node.js remain the most popular languages among Lambda users, which [continues a trend we identified in earlier reports](#). These languages were two of the earliest to be supported by Lambda, and they have garnered a large and active following in the serverless community. Typically, when organizations begin using Lambda, they start with Python and Node.js because of their ease of use and wide breadth of supported libraries, plugins, and learning materials. Then, once teams have grown more familiar with Lambda and the benefits of serverless technologies, they are more likely to migrate existing workloads that are not written in either Python or Node.js. This process has led to increased adoption of languages like Go and Java in Lambda functions, which are both used by more than 30 percent of Lambda organizations.

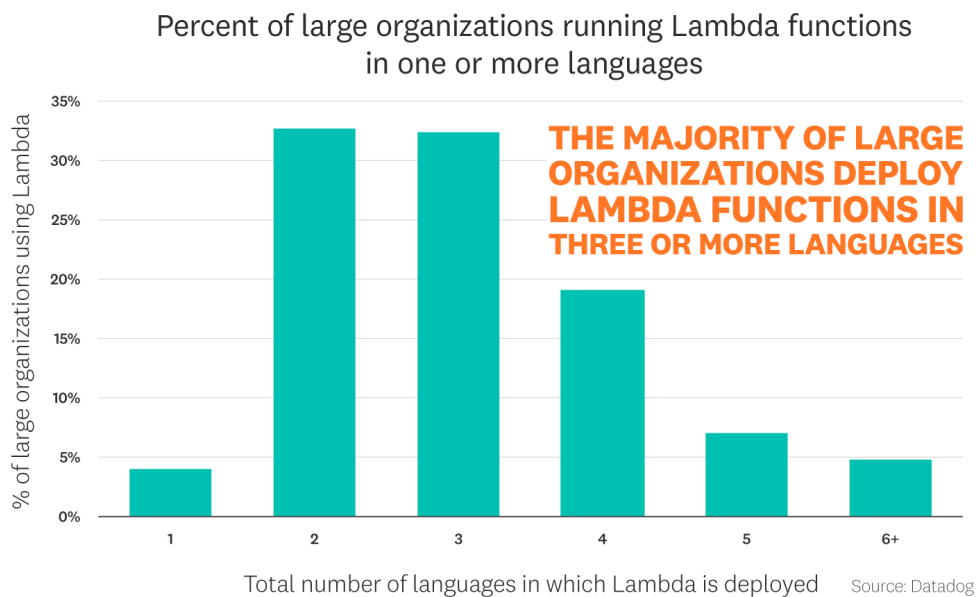
Top languages used in Lambda functions





## Over 60 percent of large organizations have deployed Lambda functions in at least three languages

Though Python and Node.js remain the most popular Lambda runtimes, 63 percent of large organizations have deployed Lambda functions in three or more runtimes.



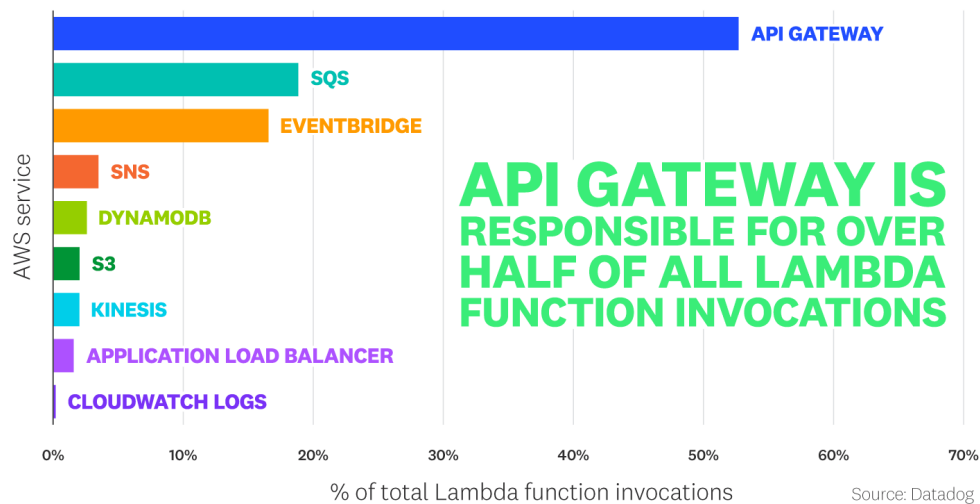
This trend illustrates that there's no one-size-fits-all approach to building serverless applications. For instance, teams may prefer to write serverless functions in a runtime that their engineers are already familiar with, such as Java. But while a large number of organizations have some Java functions, very few rely on them exclusively. This suggests that teams are opting for different runtimes depending on the use case. For example, Go's built-in goroutines make it a good fit for workloads that can be optimized with concurrency, such as data compression and video streaming.



## API Gateway and SQS are the AWS technologies that invoke Lambda functions most frequently

Lambda is AWS's most popular serverless offering, but Lambda functions rarely run in isolation. Instead, they typically execute small chunks of code in response to events emitted by a wide range of AWS services. The breadth of technologies invoking Lambda functions illustrates the extent to which organizations are using Lambda in [different ways](#) to support the unique needs of their organizations.

AWS services invoking Lambda functions



Note: This graph does not include Lambda function invocations from AWS AppSync or Step Functions.

For instance, API Gateway, a fully managed service that helps create and secure APIs, is responsible for more than half of our customers' Lambda function invocations. This fact reflects its importance to many serverless environments, in which it receives API calls, triggers Lambda functions, and then returns the response.

The second most common AWS service to invoke Lambda functions is Amazon Simple Queue Service (SQS), which manages the sending and receiving of messages between the components of your application. SQS is often used to decouple microservices and implement asynchronous communication between them, which can help improve performance in complex serverless applications. SQS is easy to configure and scale, and it also includes a fault tolerance feature called Dead Letter Queues, which handle retry logic for failed messages.

"Datadog's 2022 State of Serverless report shows not only that Lambda remains popular, but that organizations are using it alongside more and more AWS services to build highly scalable applications. It will be exciting to see how developers continue to leverage the entire AWS serverless ecosystem to support their unique business needs."

**FILIP PYREK**

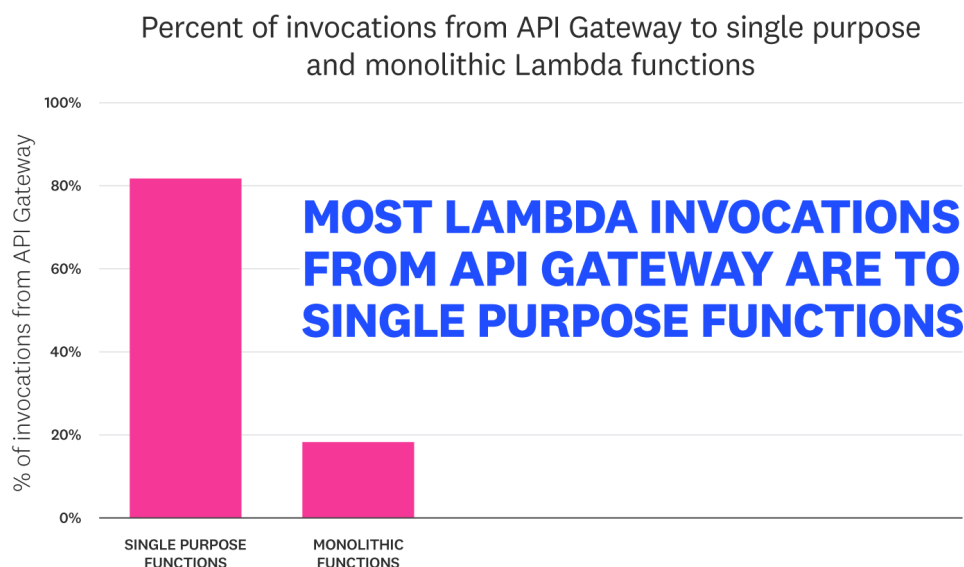
AWS SERVERLESS HERO, PURPLE TECHNOLOGY



## 80 percent of Lambda invocations from API Gateway are to single purpose functions

There are two common design patterns for serving APIs from Lambda functions: monolithic functions (also known as “mono-Lambdas”) and single purpose functions. Mono-Lambdas serve multiple HTTP endpoints and contain internal routing logic to execute several different types of tasks, whereas single purpose functions respond to only a single HTTP method and endpoint.

The vast majority of Lambda invocations by API Gateway are to single purpose functions, which are used by more than 60 percent of our customers. This data reflects the many key benefits of single purpose functions. For instance, single purpose functions are isolated from one another, which allows them to be debugged and deployed independently. Additionally, the single purpose approach enables teams to keep their applications secure by assigning one IAM role per function, which is consistent with [the best practices defined in AWS’s Well-Architected Framework](#). Finally, single purpose functions may have shorter cold start times than mono-Lambdas because of their smaller package size.



Source: Datadog

“Single purpose Lambda functions are a crucial component of modern, serverless microservice architectures. Not only can they be developed, debugged, and deployed independently of one another, but they enable teams to fine-tune resources and apply tightly-scoped IAM permissions per function. This helps to optimize performance, save costs, and increase application security.”

**JEREMY DALY**

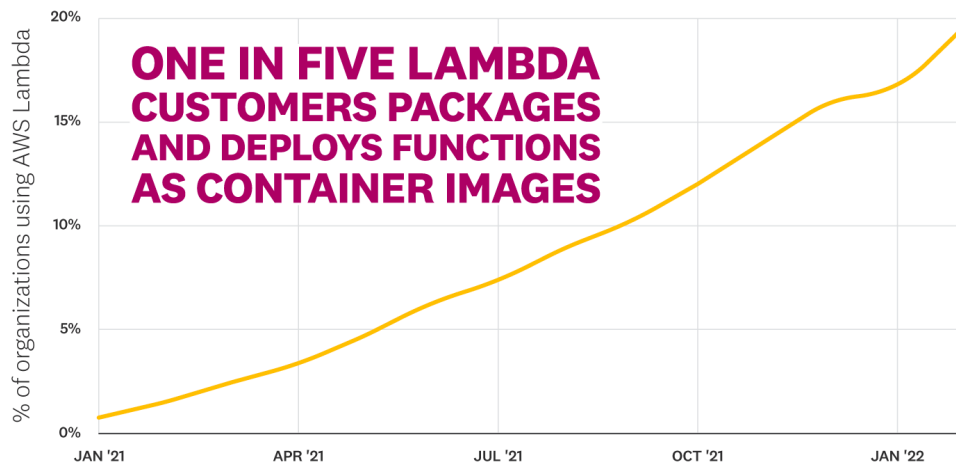
AWS SERVERLESS HERO, SERVERLESS INC.



## One in five Lambda users is deploying functions as container images

When Lambda functions were first introduced, the only deployment packages AWS supported were .zip files containing all of your function's code and dependencies. In 2020, however, [AWS added support for packaging and deploying Lambda functions as Docker container images](#). Since then, this method for deploying Lambda functions has become increasingly popular, with one in five Lambda customers using it today. This fact points to a growing and widespread interest in the combined benefits of serverless and containerized technologies—as well as the increasingly blurry line between them.

Percent of organizations deploying Lambda functions as container images over the last year



Source: Datadog

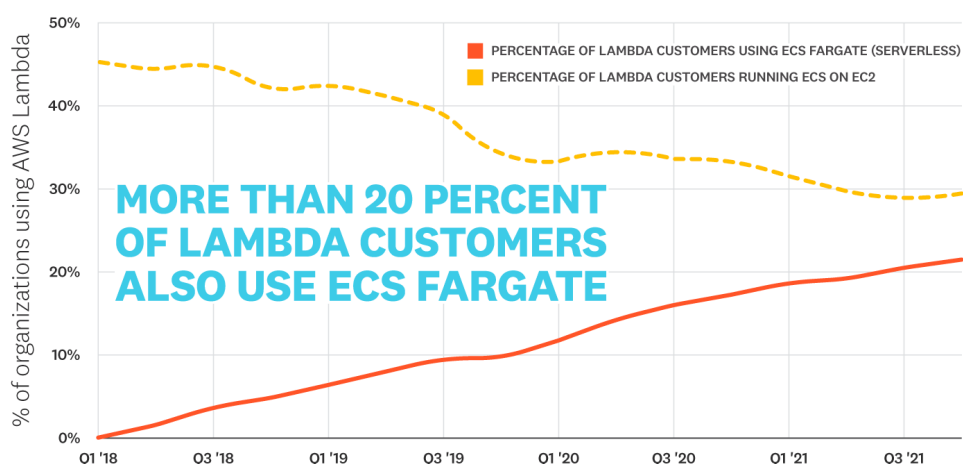
Deploying Lambda functions as container images can have a number of benefits. For example, .zip files have a size limit of 250 MB, whereas container images can be up to 10 GB. This significantly larger size limit allows organizations to leverage dependency-heavy libraries such as NumPy and PyTorch, which support data analytics and machine learning tasks. Additionally, packaging Lambda functions as container images makes it easier for organizations with existing Docker-based deployments and CI/CD pipelines to integrate serverless solutions. This ability to seamlessly incorporate serverless functions into existing workflows can save teams a significant amount of time and boost their productivity.



## More than 20 percent of Lambda customers are also using ECS Fargate

AWS customers have benefited greatly from their adoption of Lambda functions, which remain extremely popular. This success has led Lambda users to seek new ways to expand their serverless footprint. Most notably, we found that more than 20 percent of Lambda customers are also using [ECS Fargate](#) to launch containers without managing and provisioning EC2 instances.

Percent of Lambda customers who have also adopted ECS Fargate



Source: Datadog

This fact indicates that organizations are increasingly committed to serverless—and points to a deep-seated belief in the ability of serverless technology to optimize their workloads and operations. Fargate offers an appealing way to act on this commitment because it enables teams that are running applications with either ECS or EKS to easily migrate their codebase without heavily refactoring it. Fargate also gives teams granular control over resource allocation, making it well-suited for resource intensive workloads such as batch processing and machine learning jobs.

“Datadog’s 2022 The State of Serverless report continues to highlight the maturity of the ecosystem, and the breadth of business challenges that developers are able to address with serverless architectures. We’re excited to see organizations benefit from the agility, elasticity, and cost efficiency of adopting serverless technologies like AWS Lambda and Fargate, and the new paths available to them.”

**AJAY NAIR**

DIRECTOR, EXPERIENCE, AWS LAMBDA

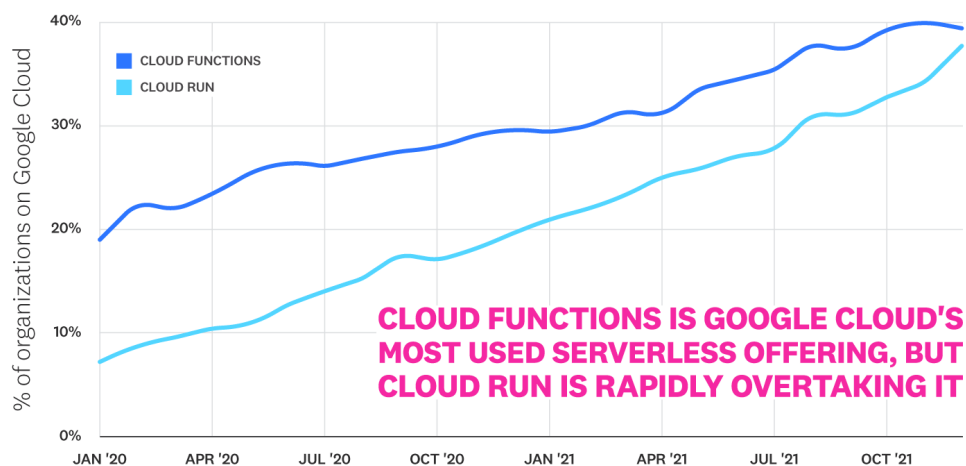




## Google Cloud Run is the fastest growing method for deploying serverless applications in Google Cloud

Nearly 40 percent of Datadog customers operating in Google Cloud have adopted Google Cloud Functions, making it the most popular serverless offering in that cloud. However, this level of adoption is only about 3 percent above that of Google Cloud Run, Google Cloud's serverless container product. This finding suggests that when it comes to serverless, an increasing number of Google Cloud users are seizing the opportunity to launch containerized applications that don't require infrastructure management.

Most commonly used serverless offerings in Google Cloud



Source: Datadog

There are several reasons why an organization may adopt Cloud Run. Whereas traditional FaaS products like Cloud Functions support only a handful of languages and execute one request at a time, Cloud Run allows users to write code in any language and can be configured to run multiple concurrent requests (in fact, all Datadog customers using Cloud Run regularly run multiple concurrent requests). This concurrency support can improve performance by reducing latency, and is especially useful for RESTful APIs with large I/O-bound workloads. Additionally, organizations can upload existing container images to Google Cloud's Artifact Registry and deploy them as microservices on Cloud Run, which simplifies the migration process. In contrast, migrating legacy code to serverless functions requires you to break down your existing application into separate services and build additional handlers that respond to specific events, which is significantly more complex.

"At Google Cloud, we're seeing serverless rapidly evolve beyond the original FaaS model toward more portable container-based applications. Cloud Run provides a fully managed, pre-provisioned platform that enables accelerated productivity and allows customers to scale rapidly without creating clusters. What's more, any containerized applications created with Cloud Run are generally portable to Kubernetes-based platforms and reduce lock-in. Datadog's latest research validates the new levels of enterprise maturity serverless delivers with a growing number of features that help increase software supply chain integrity, such as binary authorization container attestations, integration with managed secrets, and support for workload identity best practices."

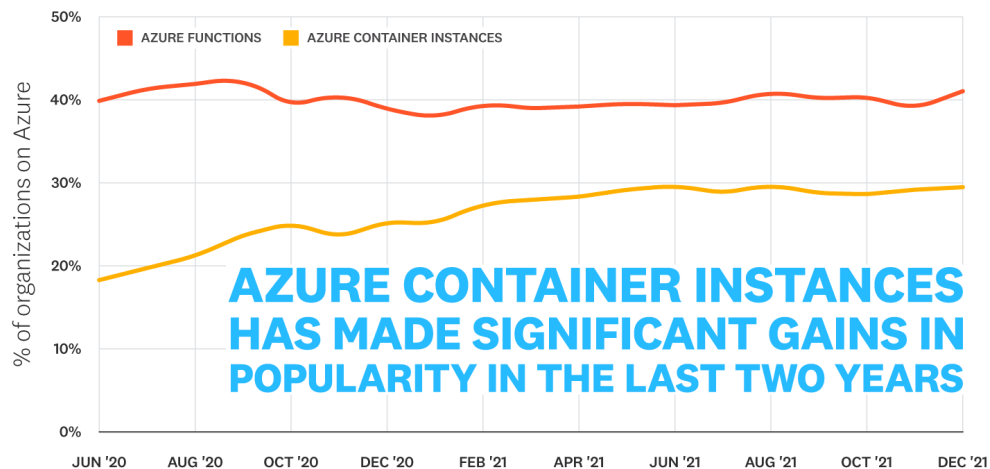
**SAGAR RANDIVE**  
PRODUCT MANAGER, GOOGLE CLOUD SERVERLESS



## Azure Functions is Azure's most popular serverless offering, but adoption of Azure Container Instances is growing fast

Azure's FaaS product, Azure Functions, is its most popular serverless offering, and it is used by more than 40 percent of Azure customers. However, Azure Container Instances (ACI), which enables organizations to run fully managed serverless containers, has seen a significant increase in adoption, and it is used by nearly 30 percent of Azure customers today. This fact echoes similar trends that we identified within Google Cloud and AWS—namely, that organizations are moving beyond the traditional FaaS paradigm and are using serverless to launch containerized workloads. We also expect to see a further increase in the adoption of other Azure serverless container technologies such as Azure Container Apps, which enables teams to build and deploy entire containerized applications in managed serverless environments.

Percent of Azure customers who have adopted serverless



Source: Datadog

One of the key features of ACI is that it enables you to get containers up and running in seconds without worrying about infrastructure management. And, if you're already using Azure Kubernetes Service (AKS), you can integrate it with ACI to automatically provision enough capacity for typical workloads—and then dynamically scale pods in a cluster during traffic spikes.

# Methodology

---

## POPULATION

For this report, we compiled usage data from thousands of companies in Datadog's customer base. But while Datadog customers cover the spectrum of company size and industry, they do share some common traits. First, they tend to be serious about software infrastructure and application performance. They also skew toward adoption of cloud platforms and services more than the general population. All the results in this article are biased by the fact that the data comes from our customer base, a large but imperfect sample of the entire global market.

---

## FACT #1

In order to determine what percentage of organizations have adopted serverless in each cloud, we had to first define "serverless compute," and then define what makes an organization a customer of a particular cloud. Our definition of serverless compute includes the following technologies:

**AWS:** AWS Lambda, AWS App Runner, ECS Fargate, EKS Fargate

**Azure:** Azure Functions, AKS running on Azure Container Instances

**Google Cloud:** Google Cloud Functions, Google App Engine, Google Cloud Run

We have not included Azure Container Apps and GKE Autopilot, which were released too recently to impact our report.

We consider an organization to be a customer of a given cloud if they run at least five hosts per month in that particular cloud. They are also considered a customer of a cloud if they run at least five functions or one serverless product per month in that cloud. Some organizations meet both sets of criteria, whereas others meet only one. For the purposes of this fact, organizations that met the latter set of criteria as of December 2021 are considered to have adopted serverless compute.

---

## FACT #3

When analyzing language usage in Lambda functions among large organizations, we defined large organizations as those that run at least 100 distinct functions per month. This figure represents approximately 33 percent of all organizations we considered. Our definition of a "runtime language" includes an aggregate of all versions of that language. The graph does not include any custom Lambda runtimes.

---

## FACT #4

In order to determine which technologies invoke Lambda functions most often, we analyzed APM trace data across all instrumented Lambda functions over a two week period, which is the default retention period for traces. We are currently unable to identify invocations from some technologies, such as AWS AppSync and Step Functions.

---

## FACT #5

We used the same methodology from fact 4 in order to determine what percentages of invocations from API Gateway are to single purpose and monolithic Lambda functions. We defined mono-Lambdas as those that are configured with a generic proxy resource, which enables the function to be called with any HTTP method, headers, query string parameters, and payload. We defined single purpose functions as those that trigger in response to a single route of a REST API. For more information, see the [API Gateway documentation](#).

---

## FACT #6

We consider an organization to be actively using Lambda if they monitored at least 20 more functions with Datadog in February 2022 than they did in February 2021.

---

## FACT #7

For this fact, we defined an ECS organization as one that was monitoring at least one ECS Fargate host with Datadog in a given month from January 2018 to December 2021.