Techstrong

Simplifying IT Operations and Boosting Developer Productivity with Serverless

Sponsored by aws

This eBook is sponsored by AWS. All opinions, writing, and recommendations are Techstrong's own.

One of the primary goals of a DevOps or platform engineering team is to enable developers to quickly build and deploy applications with the least amount of friction possible. Achieving that goal is often challenging. The underlying IT infrastructure used to run these applications historically has been too complex to manage. However, with the rise of serverless compute, infrastructure management tasks such as server provisioning, control plane upgrades, patching, scaling and maintenance are shifted to cloud providers. This reduces the cognitive load for IT teams allowing them to improve the organization's operational posture and focus on building platforms that improve developer productivity.

There are two programming paradigms through which developers can leverage serverless compute. The first is based on a functional programming model that makes it easier to embed a small piece of code within an application that enables it to invoke IT infrastructure as needed dynamically. Once a task is completed, that IT infrastructure automatically spins back down to reduce the cost of running an application. This approach reduces the total amount of infrastructure required to handle spikes in demand for additional compute resources that might be needed to process analytics data, for example.

The second is based on serverless containers, which enable IT teams to build more long-running or high-load applications that similarly invoke resources on demand. However, instead of using functions to build an application, containers provide developers with a familiar software artifact for building and deploying longer-running applications.

61% **of organizations are already leveraging serverless technologies.**

56%

of those said they either just invested in serverless for the first time or will be expanding their existing investments in the next 24 months.

Source: futurumgroup.com/insights/observability-analyst-insights-on-new-global-study

Both allow developers to consume resources on a pay-as-you-go basis, ensuring they only pay for the resources consumed. This also means infrastructure resources are not pre-provisioned, avoiding the typical upfront challenges of sizing, capacity planning and the need to overprovision resources to meet peak capacity workloads and future growth needs.

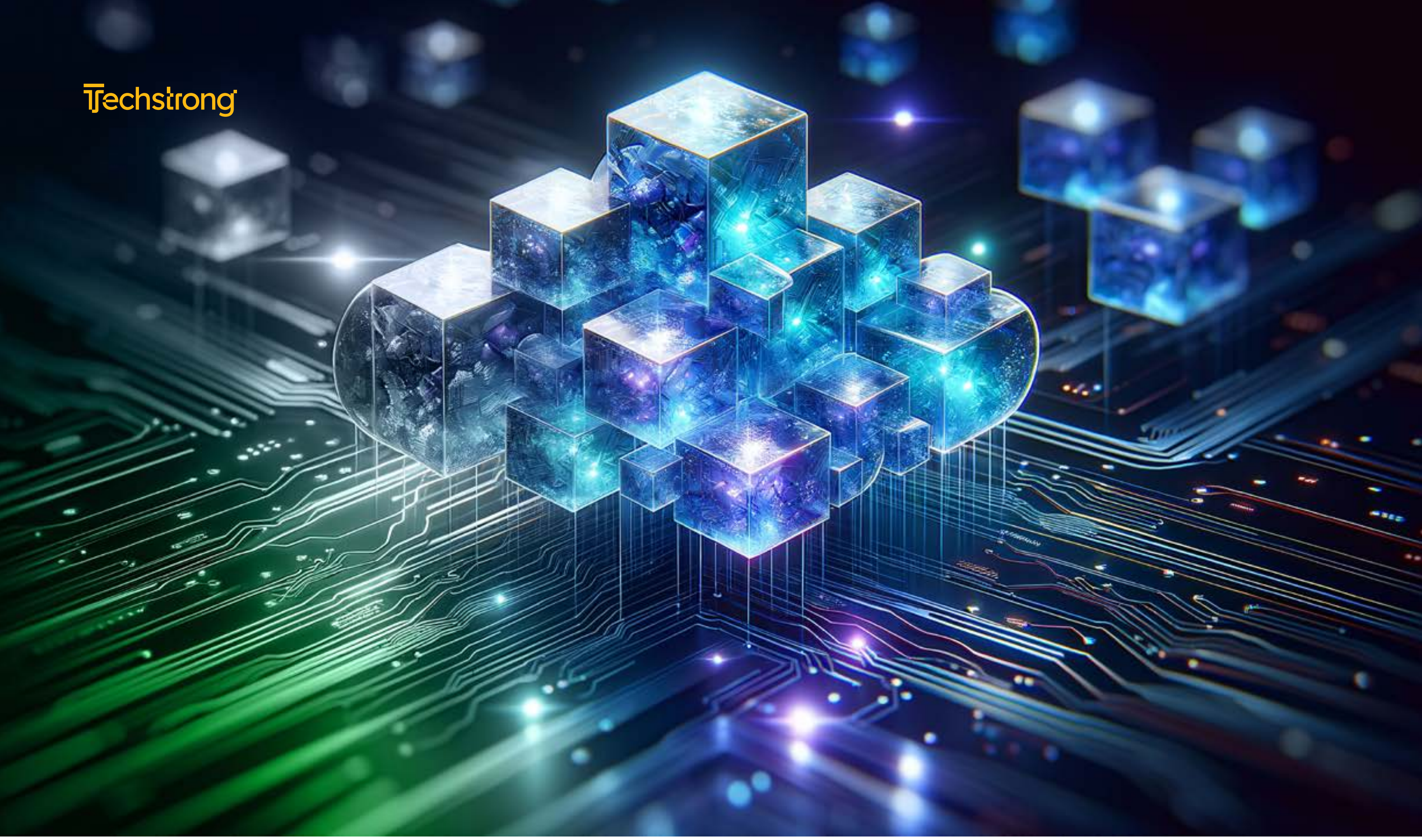
“The choice of tech stack fundamentally shapes the development, deployment, and management of applications,” said Paul Nashawaty, practice leader and lead principal analyst for The Futurum Group, a global technology research and advisory firm. “Organizations are looking to accelerate cycles from development to release to reduce overhead.”

Today, serverless compute adoption is increasing rapidly as organizations build and deploy modern applications. According to a report from Datadog, a provider of an observability platform, more than 70% of AWS customers in 2023 are using one or more serverless platforms.

Defining the Serverless Operating Model

Serverless is an operating model that enables developers to build, deploy, and run application code on dynamically allocated and managed cloud resources without the need to provision, scale or maintain cloud servers and infrastructure. The cloud provider performs server and underlying infrastructure management, patching, upgrades, security, and operations to free developers to write business logic.

Serverless enables organizations to transition to a self-service model for developers, often through internal developer portals (IDPs) provided by



platform engineering. In the book *Software Architecture Patterns for Serverless Systems*, author John Gilbert describes how serverless increases development and software delivery velocity:

"Serverless is self-service. We could offload the central teams to the cloud provider. We did not need to share resources. Teams could provision resources at will. These resources were highly available and highly observable. Teams could get up and running with new technology quickly. We could safely go into production without the learning curve."

Serverless lessens the burden on platform engineering, site reliability engineering (SRE) and DevOps teams, which are focused on automating workflows, standardizing and securing platforms, providing guardrails and tools to developers, and increasing applications' resiliency, availability and scalability. IT teams spend less time managing the "plumbing" of servers and infrastructure and dedicate more cycles to writing code that satisfies business requirements. This is why serverless has become a logical option for IT teams, who need to enable building, deploying and management of more applications than ever.

There still will be instances in which a DevOps or platform engineering team may need to manage the underlying IT platforms. But, with serverless, a much higher percentage of applications will take advantage of IT infrastructure that dynamically scales and can be managed autonomously, eliminating the need for application development teams to get bogged down by the intricacies of infrastructure management.

Building with Serverless Compute

Applications running on serverless compute platforms can use functions or containers to invoke infrastructure resources in any combination as they see fit.

Serverless functions are typically well suited for event-driven or synchronous web/request-response applications accessing external services. When an event occurs, serverless compute automatically runs the corresponding function. Functions are run without having to provision servers and can scale automatically to hundreds of thousands of instances concurrently in real time to support high application demands. When instances of that function are no longer needed, resources are released until the next event requires instances of that function.

Serverless containers are best suited for applications that need to run continuously or for longer periods, consistently get high traffic, or as a first step in a workload migrating to the cloud. This is an ideal solution for running containerized applications while getting the benefits of serverless compute. Many containerized applications follow the traditional programming approach, where applications have long-running processes with a compute layer dedicated to the process. Most existing applications follow this model, which many developers are familiar with, and this approach is used when you want to run a new or existing service-based application developed using a familiar programming approach.

A list of cloud workloads that work well with serverless functions and serverless containers are available in the section **COMMON SERVERLESS WORKLOADS**.

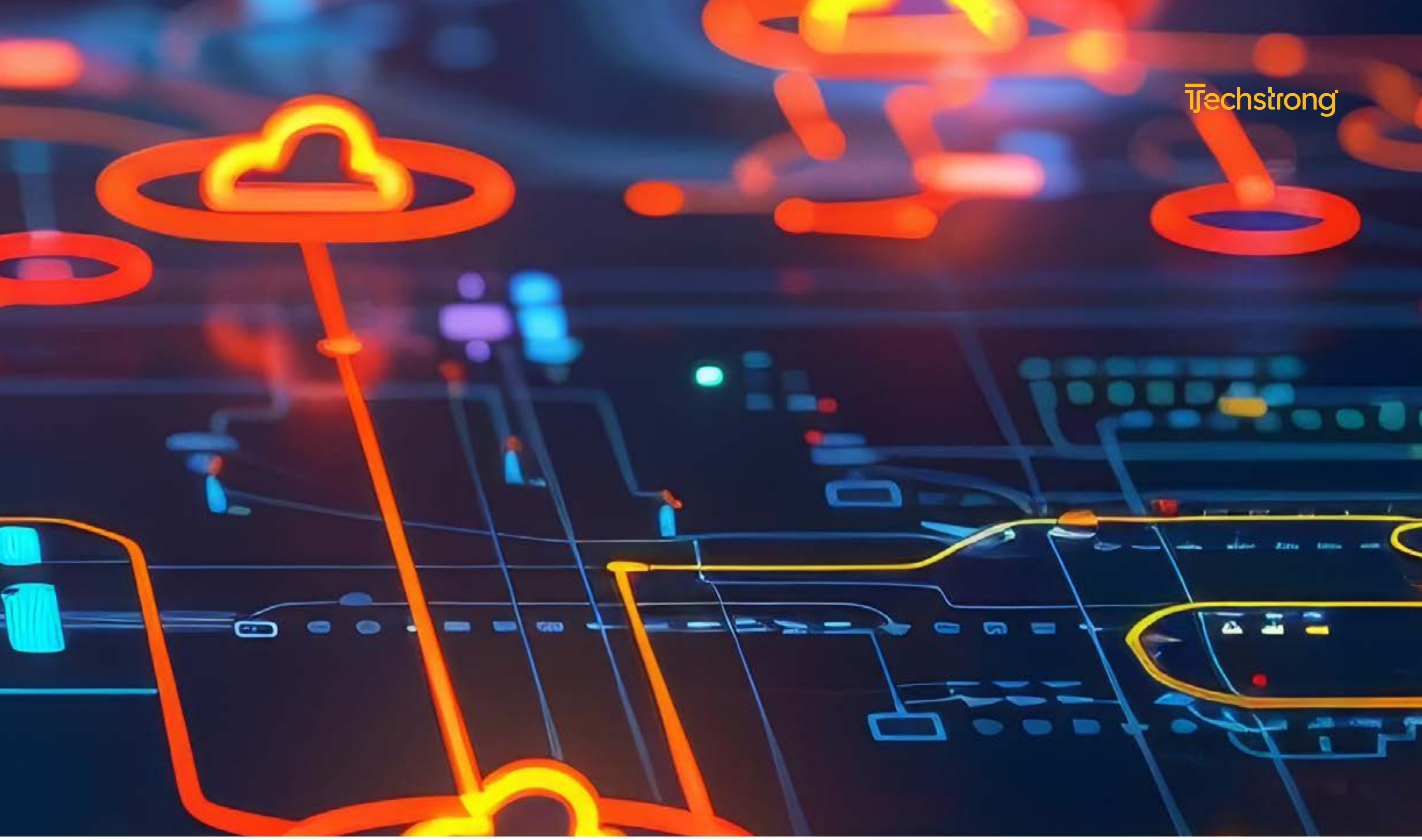
Serverless compute is well-suited to building event-driven architectures (EDA), a commonly used architectural style in which loosely coupled components interact through asynchronous event communication. To build EDAs, an event bus is employed to ingest events from various sources within an application, SaaS applications, cloud services and external services. Event buses are particularly effective when applications requiring near-real-time or real-time processing of events. The event bus ingests events, performs required transformations and delivers the event to any of the event subscribers. Because events are asynchronous, the event producers are unaware of any activity of downstream subscribers. This makes them unaware of any failures, ensuring high fault tolerance and scalability. New event subscribers can be added anytime, making serverless very effective at rapidly extending the application's capabilities by delivering new features.

Serverless Compute Benefits

Serverless compute helps DevOps and platform teams meet their goals of improving developer productivity and operational simplicity. It provides several benefits, including reduced operational burden, reliable infrastructure availability, improved security posture, and optimized costs. Let's look at each of these benefits in more depth:

Reduced operational overhead: Serverless compute reduces the need to manage the underlying infrastructure, including server provisioning, control plane upgrades, patching, scaling and maintenance—which can be time-consuming, resource-intensive, and error-prone for platform teams. With serverless compute, platform teams can set guardrails and compliance checks that meet the organization's requirements but rely on the cloud provider's services with managed operations to take advantage of best practices and expertise to improve performance, scalability, availability and security. This reduces the cognitive load for platform teams, allowing them to focus on building platforms that improve developer productivity and the organization's operational posture.

Improved infrastructure availability and scalability: Serverless compute automatically scales resources up or down in response to workload changes, ensures infrastructure is immediately available for latency-sensitive workloads, and reduces costs by cutting idle capacity. When using serverless functions, a certain number of instances can be provisioned based on historical data and traffic predictions, ensuring that they are initialized and warm, and able to instantly respond to requests in double-digit milliseconds, minimizing latency and mitigating the impact of sudden traffic spikes. Since serverless compute scales automatically and quickly,



its performance remains consistent as the event frequency increases. Cloud providers distribute serverless functions and containers across multiple cloud centers, reducing the risk of downtime and ensuring that applications remain accessible even in the face of hardware failures, network issues and other disruptions.

Improved security posture: With serverless, cloud providers take on more of the shared responsibility for security by taking on the operational burden on OS patches and runtime updates. Serverless compute also offers built-in isolation and ephemeral compute to reduce the attack surface. It simplifies building loosely coupled microservices architectures, allowing for resource isolation when compromised to minimize the impact on the broader system. Serverless compute typically integrates with security services and observability tools out-of-the-box to support end-to-end security control across multiple environments and resources throughout the development cycle, helping to strengthen overall security posture while accelerating innovation.

Reduced costs: Serverless introduces a significant cost advantage through its pay-for-value pricing model. This approach contrasts sharply with traditional compute models that require users to provision and pay for server capacity regardless of actual usage. Additionally, serverless compute further reduces the total cost of ownership (TCO) by eliminating ongoing infrastructure and software maintenance, streamlining operations and allowing development teams to focus on innovation. According to a recent Deloitte report, customers can reduce their TCO by 48% when using serverless on AWS. Serverless compute services on AWS integrate with tools that provide visibility compute costs allowing development teams to make educated compute choices for a specific workload. They also integrate with services that help identify and remediate inefficient configurations, tag resources to visualize costs, identify opportunities for savings, and provide granular per workload billing.

Common Serverless Workloads

Serverless compute and integration services can be used for several cloud workloads, especially those that require on-demand, flexible compute resources where workloads are variable, intermittent or unpredictable. The following are some examples of cloud workloads that work well with serverless:

Data Processing: Serverless compute is particularly well suited to bursty workloads like batch and real-time data processing. This includes processing large amounts of data, analytics and resource-intensive ETL workloads such as map-reduction functionality.

File processing: Serverless is ideally suited for scenarios requiring the scalable processing of files upon arrival and storage. This enables efficient data processing or the use of AI/ML.

API Backends: Serverless compute is often employed for the backends of web and mobile applications and communications between microservices. It can handle API request variability by scaling up or down automatically to match the request load.

Web Applications: Serverless containers are also well-suited for serving web pages or providing RESTful APIs, where each request is independent and state is managed locally.

Automation: Scripts run on cron jobs or in response to events for one-off processing, file conversions and automated system backups can be offloaded to serverless functions. These tasks don't require constant compute resources but need to scale on demand.

Legacy App Modernization: Modernizing traditional or monolithic applications requires breaking them into smaller microservices. These decoupled microservices can be effectively developed and deployed utilizing serverless, which supports incremental application modernization.

AI/ML: Serverless can be used to process data to train Large Language Models (LLMs) and Foundational Models (FMs). Serverless is also effective for managing generative AI data and streamlining responses to prompts back to the requesting user or API.

SaaS Integration: Serverless compute effectively integrates workflow automation between internal systems and multiple SaaS applications.

IoT Applications: Serverless compute is well suited to processing large volumes of data from IoT devices, handling the workload growth as the number of devices increases.

Event-Driven Applications: As noted earlier, serverless compute is especially effective for applications that need to react to events seamlessly in near-real time.

Summary

The appeal of serverless compute stems from offloading infrastructure management to serverless cloud services and enjoying the benefits of scale, agility, security, and cost that cloud providers offer. The significance of these benefits cannot be overstated, as they enable organizations to become more agile and responsive in today's rapidly evolving competitive marketplace. Serverless compute doesn't just optimize resources; it catalyzes innovation by freeing developers to do what they do best—create. By embracing serverless, developers can redirect their focus from the tedium of managing servers and infrastructure to the more innovative aspects of rapidly crafting software that delivers business value.

Serverless compute technologies allow future-focused organizations to utilize cutting-edge solutions to run code, manage data, and integrate applications in a way that reduces DevOps toil and improves overall developer productivity. These technologies feature automatic scaling, built-in high availability, and a pay-for-use billing model to increase agility and optimize costs. These technologies also eliminate infrastructure management tasks like capacity provisioning and patching so you can focus on writing code that serves your customers.

Serverless compute offers a pathway to a more efficient and productive future in modern cloud application development. The architectural simplicity of stateless design, the agility afforded by event-driven structures and the operational ease of managed services create an environment where developers and businesses can rapidly deliver value to their customers.

This eBook is sponsored by AWS. All opinions, writing, and recommendations are Techstrong's own.

AWS Serverless Compute Offerings

AWS Lambda is a serverless compute service that runs code as highly available, scalable, secure, resilient functions without the need to manage the underlying infrastructure. It abstracts the underlying infrastructure and allows builders to focus on application development.

AWS pioneered the serverless computing space with the launch of AWS Lambda in 2014. AWS Lambda was a revolutionary idea — run code without thinking about servers – and it has transformed and accelerated the way applications are built. This concept has since expanded, with AWS leading the charge, to encompass a wide range of services that support the serverless architecture.

Amazon Elastic Container Service (ECS) with AWS Fargate serverless compute makes it easy to deploy, manage and scale containerized applications while removing the burden of provisioning and managing servers, capacity planning, or evaluating how to isolate container workloads for security

Techstrong

Sponsored by aws

This eBook is sponsored by AWS. All opinions, writing, and recommendations are Techstrong's own.