aws marketplace

Serverless

Reviews, tips, and advice from real users






Powered by PeerSpot

Contents

Product Recap

Serverless

Serverless Recap

Serverless revolutionizes application architecture by eliminating the need for server management, offering a scalable, cost-efficient approach for modern development needs. It enables developers to focus on writing code without the complexities of infrastructure handling.

Originally designed for enhancing agility, Serverless provides on-demand function execution, ensuring seamless scalability and rapid deployment. As a back-end architecture, it allows businesses to execute code in response to events and automatically manage the required resources, optimizing resource usage and minimizing idle capacity. This approach abstracts server provisioning, allowing developers to streamline workflow and focus on creating innovative solutions without dealing with infrastructure concerns.

What are Serverless's key features?

- Event-Driven Execution: Triggers functions in response to specific events, automating execution based on user actions or data changes.
- Dynamic Scalability: Adjusts resources automatically in real-time to handle varying workloads without manual intervention.
- Pay-As-You-Go: Charges only for the execution time and resources used, optimizing costs and ensuring economical resource use.
- Automatic Resource Management: Handles server provisioning, scaling, and maintenance, freeing developers from managing infrastructure intricacies.

What benefits should users expect in reviews?

- Increased Developer Productivity: Allows developers to focus on code, leading to faster feature releases and quicker innovation cycles.
- Cost Efficiency: Operates on a usage-based billing model, reducing overhead costs and financial wastage due to idle resources.
- Enhanced Scalability: Seamlessly accommodates fluctuating demand, providing reliability and stable performance.
- Reduced Time-to-Market: Facilitates rapid deployment and iteration, helping teams release applications faster.

Serverless finds application across industries such as e-commerce, where it handles app backends, and finance, for managing real-time data processing tasks. Its flexibility supports a diverse range of enterprise needs, including automated data collection in retail and transaction processing in banking.

Valuable Features

Excerpts from real customer reviews on PeerSpot:

"We have found that Serverless works extremely well for our use case because it allows our small team to deploy and maintain over 100 functions without significant infrastructure overhead."

Yash Patel

Software Developer at BISC (Bhaskar Jarian Institute for Space Application and Geoinformatics)

"Serverless significantly boosts ROI by lowering infrastructure expenses by 40% to 80%, enhancing development speed by 30% to 60%, and minimizing maintenance requirements, translating to faster delivery and decreased operational costs."

D Cs

Software Engineer at Cypherox Technologies pvt ltd

"In terms of time or cost saved compared to before using Serverless, I save approximately 60 percent of my development time because everything is very lightweight and gives me the freedom to work within Serverless."

Sachith Delaga

Lead Software Engineer

"Serverless has positively impacted my organization by shifting our focus from infrastructure management to pure product delivery."

Hussain Gagan

FullStack Developer at EnactOn Technologies

"I can share specific metrics indicating Serverless's positive impact: I achieved 100% uptime, an impressive feat compared to traditional servers that often experience downtime during peak usage."

Hamza Sharif

Cloud Engineer at a consultancy with 11-50 employees

"Serverless has positively impacted my organization as it has become the de facto standard for any POCs I have been doing, unless something specific is required that demands an on-demand instance."

Mukesh Srivastav

Senior Software Engineer at Tech9

"Serverless impacts my organization positively in many ways by enabling us to easily debug issues when any pipelines break; we can get errors, debug them, and address issues."

ArunKumar30

Working For M Bank at a financial services firm with 51-200 employees

What users had to say about valuable features:

“There is a huge impact as my traffic gets auto-adjusted. I do not have to worry about whether my server is capable of handling the traffic or not. Serverless servers are much more capable. I do not have to bear the cost burden. I just need to pay for whatever I am using.

“Serverless has definitely improved cost savings and there are fewer order failures due to high traffic..”

Amar-Kumar
Technical Lead at a tech services company with 501-1,000 employees

Read full review

"Serverless works smoothly with services like Amazon Web Services, S3, DynamoDB, API Gateways, Google Cloud, and Microsoft Azure. Serverless provides no server management, pay-as-you-go pricing, automatic scaling, fast deployment, event-driven execution, high availability, and easy integration with cloud services. It provides automatic scaling, pay-per-use pricing, and offers fast, cost-effective, and highly scalable operations.

Third-party API integration is an important feature, as you can connect the app via APIs, payment gateways, webhooks, CRM integration, and the WhatsApp API. Chatbots and automation bots handle user queries automatically for customer support and lead generation.."

D Cs

Software Engineer at Cypherox Technologies pvt ltd

Read full review

"The best features Serverless offers beyond image processing include building event-driven APIs and cron-like automations. For instance, I set up scheduled Lambda functions to handle daily database cleanup and report generation. For me, the biggest advantage is the automatic scaling and the pay-per-execution model, allowing us to handle massive traffic spikes without manual intervention.

"During high traffic periods, I found that automatic scaling has helped us immensely. We had a major marketing campaign launch last year that drove a sudden 10x spike in traffic to our platform, and because our backend was built on Serverless functions, the infrastructure scaled out instantly to handle the concurrent requests without me having to provision a single extra server or worry about downtime.

"Serverless has positively impacted my organization by shifting our focus from infrastructure management to pure product delivery. By offloading the operational overhead to the cloud provider, my team has been able to cut our time to market for new features by nearly 30%.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"I believe the best features Serverless offers are the very quick ability that enables individuals to quickly make calls to their back-end or to quickly make calls to their services. Additionally, Serverless is very useful when it comes to running simultaneous jobs at the same time without breaking.

"Serverless helps run simultaneous jobs. For instance, when you need to make a back-end API call, multiple people can make such calls at the same time. What happens at the Serverless back-end is it creates something similar to multiple instances or multithreading that allows each Serverless Lambda or each Serverless resource to run concurrently without affecting one another.

"It has helped a lot in saving costs because, as I mentioned initially, it makes sure services are not being used unless they are being invoked. It has really helped in making sure costs are well managed and also making sure we do not make use of resources that are not needed at a particular point in time.

"Making use of Serverless has at least helped us save 50% in cost spending on resources.

"Because I believe Serverless has had a very positive impact on myself and also on the company I work for, especially on the cost side. It is very cost-effective and has helped us to save a lot, I believe up to 50% on cost savings and also has helped us to really save a lot of money when it comes to deploying back-end and managing back-end services.."

Daniel Asha
AI Engineer at a consultancy with 11-50 employees

Read full review

"The best feature that Serverless offers is that I do not have to manage any servers because the service providers, like AWS or other cloud providers, take full care of the servers behind the scenes, which means I do not have to manage maintenance, security, scalability, or anything about those servers. I focus my attention on application development rather than spending time on servers.

"Serverless has helped me and my team by making our workflow easier and freeing up time for other tasks. If I focus on previous projects, particularly the fintech project, which operates like a Revolut application and is based in Haiti under the name MonCash, I deployed microservices in Fargate that are highly scalable. The application supports features like adding money, sending money, transfers, and bill payments, and I needed to avoid spending time troubleshooting infrastructure because everything was Serverless, making it very easy to manage, highly durable, and secure.

"In the previous project, monitoring was done solely on AWS CloudWatch, despite not having access to servers or SSH. Still, I had monitoring capabilities for our services. For example, if a service reached 90% capacity, I could set auto-scaling limits, ensuring costs remained manageable. Integration was handled through AWS Cloud Map, managing the networking of new IPs for our microservices, which is also a Serverless service.

"Serverless has positively impacted my organization, particularly through its scalability. Developers can deploy at any time thanks to blue-green deployment available in this architecture, allowing for bug fixes or new features to be pushed into production without any downtime, which has helped not only my organization but also the fintech application MonCash, which has enjoyed uninterrupted service, meeting SLAs consistently.

"I can share specific metrics indicating Serverless's positive impact: I achieved 100% uptime, an impressive feat compared to traditional servers that often experience downtime during peak usage. With Serverless, I had 100% uptime SLA, which was excellent for my portfolio and essential for end users.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"Serverless helps me with data processing in AWS by making infrastructure deployment very easy. With a single click, it automates everything when I am working inside AWS infrastructure. The development is also very fast and easy to implement, and it is not a complex architecture compared to Spring Boot, MVC, or other infrastructures.

The best features Serverless offers for me include automated deployments, which are very smooth and interesting. As a full-stack engineer, Serverless really helps me to reduce my DevOps cost. Another valuable feature is the offline mechanism, and I have used AWS LocalStack with Serverless Offline, which is really interesting and helps me to simulate cloud infrastructure without any cost on my machine.

Automated deployments and the offline mechanism impact my workflow positively because when I configure the Serverless application in the serverless.yml file, I can configure everything, such as AWS services I have used in my application, including S3 configuration, Cognito configurations, and database configuration as separate YAML files integrated into one Serverless file. Then I just click on the NPM deploy, develop, or any staging option, and it automatically deploys to my AWS CloudFormation stack, creating the entire service.

Serverless positively impacts my organization by allowing us to work as a startup with very limited resources and costs. When we go with a Serverless infrastructure, we reduce the need for specialized resources, especially on the DevOps side, because everything becomes automated, enabling our full-stack engineers to perform that work. Reducing resources means we reduce cost as well, and it is time-saving since deployment does not take hours but rather depends on our network speed.

Serverless helps with scaling my applications as the organization grows by not restricting the inclusion of more components or modules in the Serverless applications. However, there can be some restrictions. For example, AWS S3 only supports a maximum file upload of 250 MB using Serverless. Despite a few concerns, from a Serverless point of view, integration and implementation of our logic into applications remain very easy.."

Sachith Delaga
Lead Software Engineer

Read full review

Other Solutions Considered

"Before moving to Serverless, we were running a monolithic application on standard EC2 instances. We decided to switch because scaling was manual and reactive, which led to significant downtime during traffic spikes and high operational overhead for our engineering team.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"Before choosing Serverless, I considered other solutions such as Express with AppRunners, but I found that version to be time-consuming compared to traditional Serverless with more manual interference, especially regarding deployment and DevOps. I also checked Spring Boot, but it did not match our application needs, focusing instead on AppRunners with Express as a viable alternative.."

Sachith Delaga
Lead Software Engineer

Read full review

"I migrated from using Docker containers on EC2 instances to Serverless due to numerous challenges in DevOps, including complex monitoring setups and the extensive automation needed to scale infrastructure. Serverless simplified my architecture, making it highly available and scalable without managing servers.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"Before choosing Serverless, we evaluated other options and looked into containerizing our monolith with Kubernetes on EKS. While Kubernetes offered great portability, we ultimately decided against it because the operational overhead of managing clusters did not solve our core problem of wanting to focus purely on feature development rather than infrastructure maintenance.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"I have used several solution frameworks previously, including Java Spring Boot and NestJS with EC2, among others. The decision to switch to Serverless is based on the project or company requirements. If I was working in a very large enterprise application, I would choose Java over Serverless. However, for this startup, we determined that Serverless was the most suitable framework, and I am open to switching frameworks in the future as per the architecture needs of the application.."

Sachith Delaga
Lead Software Engineer

Read full review

"We evaluated other options before choosing Serverless, such as Kubernetes, which has more control but significantly higher operational complexity. For our use case, it was overkill. Google Cloud's App Engine is similar to Lambda but had a less mature ecosystem at the time. AWS Lambda has better integration with other services we needed. Azure Functions was a viable option, but our organization was already invested in the AWS ecosystem. We considered continuing with EC2, but it did not address our scaling and cost concerns. Lambda was the clear winner for our requirements.."

Yash Patel
Software Developer at BISC (Bhaskar Jarian Institute for Space Application and Geoinformatics)

Read full review

ROI

Real user quotes about their ROI:

"I have seen a return on investment with Serverless, as moving from servers means requiring fewer cloud engineers and DevOps staff for maintenance, patches, and infrastructure management, resulting in reduced time and effort.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"We definitely saw a strong return on investment after moving to Serverless architecture. By reducing our monthly infrastructure spend by about 30%, we eliminated the idle capacity costs we were previously paying for underutilized EC2 instances.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"I have seen a return on investment. Previously, we needed one full DevOps person to handle all of that, but now with Serverless, our developers can easily and quickly get the application up and running. With Serverless, we needed fewer employees and also saved time.."

Hallie Greenfelder
Full Stack Developer

Read full review

"In terms of time or cost saved compared to before using Serverless, I save approximately 60 percent of my development time because everything is very lightweight and gives me the freedom to work within Serverless. Similarly, regarding cost, when I reduce time, I should automatically reduce cost as well. About deployment, we handle deployment more than 80 percent faster, so we do not need to have a specialized DevOps engineer as my full-stack skills cover it.."

Sachith Delaga
Lead Software Engineer

Read full review

"I do not have fixed data, but I can give approximate data regarding return on investment with Serverless. For direct cost savings, we achieved a 20% reduction in cloud spending compared to an EC2 approach. For development time, new API deployment time was reduced from one day to one to two hours. For team scaling, we maintained the same team size while handling three times more projects. New applications go live two to three weeks faster. Production issues are resolved 35% to 40% faster due to better monitoring. We eliminated 10 hours per week of infrastructure maintenance.."

Yash Patel

Software Developer at BISC (Bhaskar Jarian Institute for Space Application and Geoinformatics)

Read full review

"Return on investment is realized through significant savings in infrastructure costs since the server is not constantly running; you only pay during actual usage, typically realizing 40% to 80% lower costs. There is also a faster time to market facilitated by quicker build and deployment times, which allows products to launch sooner and start generating revenue earlier. Time savings stem from faster development cycles, as maintenance costs drop due to a lack of server patching, requiring less DevOps effort. Smaller teams are required, contributing to lower salary and operational costs, complemented by automatic scaling that prevents the need to over-invest. You do not need to acquire additional servers, thus avoiding wasted funds. A straightforward ROI scenario highlights that with a conventional setup the server's monthly cost is $3,000, in addition to $10,000 for DevOps, reaching a total of $30,000 per month. Contrastingly, with a Serverless setup, the usage cost reduces to $1,000 with minimal maintenance, leading to $1,000 to $3,000 monthly expenses. ROI calculations indicate a savings of $10,000 each month, representing a 300% to 500% improvement. Serverless significantly boosts ROI by lowering infrastructure expenses by 40% to 80%, enhancing development speed by 30% to 60%, and minimizing maintenance requirements, translating to faster delivery and decreased operational costs.."

D Cs

Software Engineer at Cypherox Technologies pvt ltd

Read full review

Use Case

“My main use case for Serverless is that I mainly worked on Node.js serverless applications for my platforms, and I have worked with different domains, spanning three or four domains with Serverless.

A specific example of how I used Serverless in one of those domains is that I mainly worked with AWS infrastructure using the AWS stack, including S3, AWS Auth, and Cognito. I use several AWS services with Serverless and Node.js..”

Sachith Delaga
Lead Software Engineer

Read full review

“I use Serverless to deploy back-end APIs and to run serverless applications, which are basically microservices.

“I make use of AWS Lambda to deploy back-end for artificial intelligence applications. For instance, one example I deployed using AWS Lambda was for the back-end of an application where the front-end calls the back-end to return data. This helps ensure that the back-end operates separately, and resources are not being used when not needed.

“I run serverless applications on AWS, and I believe the main use case is to ensure that application back-ends are not being used unless they are specifically called or unless they are specifically needed for use..”

Daniel Asha

AI Engineer at a consultancy with 11-50 employees

Read full review

"The main use case we are using is auto-scaling and cost-effectiveness. Some of our use cases involve unpredictable traffic. For example, during Eid events, I am from the QSR domain, so traffic on Eid day is not predictable. When using Serverless, it auto-scales, and I pay based on actual usage.

"In my case, I use everything on my main server for what we build, but for order processing, we are using Serverless where we do not want any hassle of server management, such as upscaling. Order processing is the key part of my application. I preferred to use Serverless for this part so that none of my customers face any problems processing orders, because if any order fails, it loses the customer's confidence or trust.

"I suggested my team use auto-scaling and Serverless for order processing and notifications, with auto-adjusting features to auto-manage traffic. For this feature, we are using Serverless.."

Amar-Kumar
Technical Lead at a tech services company with 501-1,000 employees

Read full review

"My primary use case for Serverless is handling asynchronous data processing and event-driven workflows. I typically use it to trigger background tasks like image processing or data transformation whenever a file is uploaded to S3, which keeps our main application responsive.

"In my last role, I used Serverless to address an issue where users were uploading high-resolution images that were slowing down our main site. I set up an S3 trigger that automatically invoked a Lambda function the moment a file hit the bucket, and the function resized the image into three different formats and stored them back to a separate bucket, which reduced our page load time by about 40% and significantly lowered our storage cost.

"By offloading that processing to the background, we ensured that the main application remained responsive while the images were handled asynchronously, turning a major performance bottleneck into a seamless, automated workflow for our users.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"In my previous job and in my current job, I work as a cloud engineer, where I have been working with some clients who have provisioned Serverless architecture for their business, and I provide services to those customers as a cloud engineer.

"I can give you a quick specific example of a project where I used Serverless: in my previous company, I worked on a fintech project where the services ran in Fargate, a Serverless service of AWS, deploying a microservices architecture within this Serverless framework. In my current job, I also provide support for a customer whose entire architecture is deployed in Serverless on AWS Cloud, which includes API Gateway, Lambda functions, DocumentDB, and S3 buckets; everything within this architecture is Serverless, and I provide maintenance and daily support for this project in my current job as well.

"I mainly worked on these two projects with Serverless, but I know there are other Serverless services in AWS that I have not worked with in a production environment. Thus, I can say these two are the main projects I have been involved in with Serverless architecture.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"Serverless is primarily used for creating backend services, with the most common use being REST APIs without running a full server. Each API endpoint is a function that scales automatically with traffic, such as login, signup, and payment APIs. File processing is triggered when a file is uploaded and automatically resizes, compresses, and converts the image, for example when a user uploads a profile picture. Event-driven automation runs when something happens on an event basis, eliminating the need to continuously run the server, such as sending an email after registration or triggering an SMS when an order is placed. Serverless is mainly utilized when you want a scalable backend and logic that runs only when needed to save cost and effort. It is also used for chatbots and AI integration, processing the user's message instantly and integrating with AI and APIs, such as a WhatsApp bot and support chatbot.

The API and backend play the most important roles in my focus, as you can build a backend without a server for login, signup, APIs, CRUD operations, and payment endpoints, which are the most common real-world use cases. File and image processing involves a trigger that enables resizing an image, compressing a file, or converting formats. Event-based automation includes sending an email after signup or an order confirmation and SMS for data processing. Scheduling jobs with a cron task can run daily reports, backup jobs, or data cleanup at a specific time. Chatbot and AI integration, such as AI and API integration for customer support bots, represents another use case. Microservices are also needed, breaking the app into small services like auth services and payment services.

In day-to-day use of Serverless in our software company, we mostly rely on it for forms, contact forms, e-commerce, image optimization, and notification alerts. When a user fills out a contact form, a serverless function is triggered to send an email and store the data, which removes the need to build a full backend. E-commerce automation is significant, as when an order is placed, it triggers a function to send the invoice, confirmation emails, and update the database in the CRM, which we use with WooCommerce and Shopify integrations. We utilize WordPress and Shopify websites, making this approach very beneficial. Image optimization occurs when you upload an image, as it is automatically resized and compressed, thereby improving website performance, which is useful for websites

and blogs. Notifications and alerts enable the sending of emails, SMS, push notifications, login alerts, payment success notifications, and scheduling tasks. Automatic scaling is particularly helpful in our day-to-day work.."

D Cs

Software Engineer at Cypherox Technologies pvt ltd

Read full review

Setup

The setup process involves configuring and preparing the product or service for use, which may include tasks such as installation, account creation, initial configuration, and troubleshooting any issues that may arise. Below you can find real user quotes about the setup process.

"My experience with pricing, setup cost, and licensing demonstrates that pricing is good because we only get charged for the amount of time we use and the amount of time we trigger events. Setup cost is minimal because it is easy to use. It requires only some code, which is why it is easy to set up.."

Verified user

Cloud Engineering and Automation Engineer at a tech vendor with 51-200 employees

Read full review

"My experience with pricing, setup cost, and licensing was fine; pricing is very good for minimal or moderate application code. Pricing increases with larger codebases, but for a normal moderate codebase, it is good. The setup is minimal, and licensing is also acceptable since it is handled by Amazon.."

Prashanth Dembe

Founding Engineer

Read full review

"I estimate that the development process with Serverless is now 30% to 60% faster compared to before I started using it, primarily due to faster deployment. The setup time has reduced from days to minutes; traditionally, a backend setup takes three to five days, whereas a Serverless setup takes about two to four hours, showcasing great time savings. Over time, a project that would take two months can now be achieved in just one to one and a half months.."

D Cs

Read full review

Software Engineer at Cypherox Technologies pvt ltd

"We did not purchase Serverless through the AWS Marketplace; we manage our infrastructure directly through AWS accounts using Terraform for our IAC, which gives us better control over environment configuration and deployment pipelines.."

Hussain Gagan

Read full review

FullStack Developer at EnactOn Technologies

"Serverless is not available on the AWS Marketplace; it consists of a collection of services already available on the AWS console, allowing me to use any service I need like Lambda for compute, DocumentDB for databases, S3 for file storage, and API Gateway for serverless APIs. There is no single service to purchase, and setup is straightforward, with no licensing required—however, costs remain higher than desired to attract more customers.."

Hamza Sharif

Cloud Engineer at a consultancy with 11-50 employees

Read full review

"Our experience with Serverless pricing, setup cost, and licensing is very transparent. You pay for execution time, number of requests, and data transfer. There are no hidden costs or licensing fees. The setup cost is minimal. You need an AWS account set up, but no capital investment is required. Our ROI was positive within six months. We reduced infrastructure costs by 15% to 25%. Development time was reduced by 30% to 40%. The team did not need to grow despite handling five times more applications. This is one of Serverless' biggest advantages: a low startup cost and quick ROI.."

Yash Patel

Software Developer at BISC (Bhaskar Jarian Institute for Space Application and Geoinformatics)

Read full review

Customer Service and Support

"The customer support experience has not been applicable to me, as this was done by my solutions architect, so I never interacted with the customer support.."

Verified user
Product Manager at a tech vendor with 10,001+ employees

Read full review

"I would rate customer support around 8 out of 10 because it is consistently quick, the documentation is comprehensive, and all customer support is quite responsive, so there is not much of a blocker.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"My experience with AWS support regarding Serverless is mixed; while I appreciate the interaction, I lack deep visibility into the monitoring and logging of Serverless components. When issues arise, I rely on AWS for detailed insights, but the lack of direct access can be limiting.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"AWS support for Lambda issues has been outstanding. We get a quick response time, within one to two hours, for critical issues. The engineers understand Serverless deeply, and there is comprehensive documentation for troubleshooting. We have rarely needed to escalate because the documentation is so thorough. We use an AWS Support plan and recommend at least the Business tier for production. It has been worth every penny.."

Yash Patel

Software Developer at BISC (Bhaskar Jarian Institute for Space Application and Geoinformatics)

Read full review

Other Advice

"I would tell them that if they want something quick, portable, and fast, they can make use of Serverless. However, if what they want is something that has to do with data that is needed in real time, then they should look for a different solution. I give this product a rating of 8.."

Daniel Asha
AI Engineer at a consultancy with 11-50 employees

Read full review

"If you are a startup or have any stable product and you want on-traffic payment, then you should definitely use Serverless. If you are not able to predict your traffic, then you should definitely use Serverless. For example, some days we have one hundred orders, but on a big day, we may have hundreds of thousands of orders. You cannot upscale your server from day one. You should definitely shift to Serverless. It will definitely help you reduce your costs and you can easily manage your traffic. I would rate this product as a 9 out of 10.."

Amar-Kumar
Technical Lead at a tech services company with 501-1,000 employees

Read full review

"My biggest piece of advice for others looking into using Serverless is to prioritize observability from day one because you lose visibility into the underlying infrastructure, so you need to have robust logging and distributed tracing in place immediately, or debugging becomes a nightmare.

"One final point about Serverless is that while it is incredible for scaling, I think it is crucial to be mindful of cold starts and vendor locking early on; if you design your architecture to be modular from the start, you keep your options open as the system grows. I would rate this product an 8 out of 10 overall.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"I would rate Serverless an 8 out of 10.

"I choose an 8 out of 10 because of the pricing, which should be reduced, even though a higher price makes sense due to the services provided, but AWS pricing is significantly higher than traditional servers. Additionally, access to Serverless offerings must be more accessible for all users to become the first choice for customers.

"I advise others to consider Serverless as their first option, as it saves effort and money despite the higher costs, but the reduction in maintenance and deployment costs from traditional servers is significant. My overall review rating for Serverless is 8 out of 10.."

Hamza Sharif
Cloud Engineer at a consultancy with 11-50 employees

Read full review

"Serverless as a Function-as-a-Service (FaaS) effectively runs small functions on demand, where each task is executed as a function through trigger-based activation, exemplified by an API endpoint that instigates a function. From an event-driven architectural perspective, the system reacts to events like file uploads, API calls, or successful payments, with everything executed as a function

triggered by specific activities. Adopting a microservices model divides the application into small, independent services such as auth services, payment services, and notification services, where each function qualifies as a microservice. Under a cost optimization strategy, it emphasizes economical operations without idle costs, requiring payment strictly for execution, which renders it an optimal solution for startups and low-traffic applications. Moreover, as a scalability engine, it includes built-in automatic scaling to manage unpredictable traffic without human intervention, acting as a beneficial tool to enhance developer productivity.

Serverless is optimal for our organization, making it exceptionally helpful. I give this solution a rating of 10 out of 10.."

D Cs
Software Engineer at Cypherox Technologies pvt ltd

Read full review

"Serverless affects my team's productivity and collaboration by presenting some challenges. For instance, when we work in parallel, deploying two different versions at the same time can lead to issues or conflicts, where resources may not be generated successfully. However, these challenges are manageable since we typically avoid deploying two versions at the same time in actual production development.

Serverless handles monitoring and troubleshooting effectively by integrating with CloudWatch, allowing for easier understanding of logs. Serverless also possesses extensive documentation and references, making it easy to resolve any issues related to its functionality, although logic issues might require different handling.

My advice to others looking into using Serverless is that you need to understand your requirements and ensure Serverless aligns with your solutions. It depends on your application, the cloud solution being utilized, and the services required. Throughout my experience with Lambda, I always tie it back to that. While it works for me, others might have different needs or infrastructures, hence it is crucial to have an open mindset and determine what framework truly suits you rather than sticking to one blindly, as that could lead to frustrations. I would rate my overall experience with Serverless as a seven out of ten.."

Sachith Delaga
Lead Software Engineer

Read full review

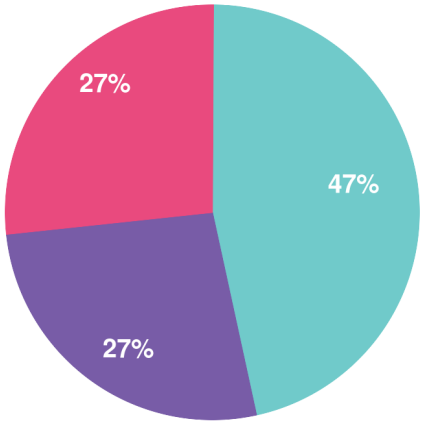
Top Industries

by visitors reading reviews

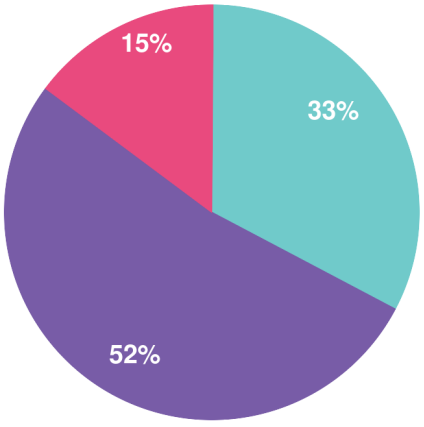
Construction Company	34%
Financial Services Firm	12%
Outsourcing Company	7%
Manufacturing Company	7%

Company Size

by reviewers



by visitors reading reviews



Large Enterprise Midsize Enterprise Small Business

About this buyer's guide

Thanks for downloading this PeerSpot report.

The summaries, overviews and recaps in this report are all based on real user feedback and reviews collected by PeerSpot's team. Every reviewer on PeerSpot has been authenticated with our triple authentication process. This is done to ensure that every review provided is an unbiased review from a real user.

Get a custom version of this report... Personalized for you!

Please note that this is a generic report based on reviews and opinions from the collective PeerSpot community. We offer a customized report of solutions recommended for you based on:

- Your industry
- Company size
- Which solutions you're already considering

The customized report will include recommendations for you based on what other people like you are using and researching.

Answer a few questions in our short wizard to get your customized report.
Get your personalized report here

About PeerSpot

PeerSpot is the leading review site for cloud, AI, and business software. We created PeerSpot to provide a trusted platform to share information about software, applications, and services. Since 2012, over 22 million people have used PeerSpot to choose the right software for their business.

PeerSpot helps tech professionals by providing:

- A list of products recommended by real users
- In-depth reviews, including pros and cons
- Specific information to help you choose the best vendor for your needs

Use PeerSpot to:

- Read and post reviews of products
- Access over 30,000 buyer's guides and comparison reports
- Request or share information about functionality, quality, and pricing

Join PeerSpot to connect with peers to help you:

- Get immediate answers to questions
- Validate vendor claims
- Exchange tips for getting the best deals with vendor

Visit PeerSpot: www.peerspot.com

PeerSpot

244 5th Avenue, Suite R-230 • New York, NY 10001

reports@peerspot.com

+1 646.328.1944