Determining the Total Cost of Ownership of Serverless Technologies when compared to Traditional Cloud

September 2019 v2

Deloitte Consulting

Authors:
Akash Tayal, Eric Lam, Diganto Choudhury, Meghan Dickerson,
Ganesh Moovera, and Gary Arora

Deloitte Industry Insight

As CIOs continue to drive cloud computing from being bleeding edge to establishing it as a mainstream technology capability within the organizations, recent forays have also focused on shifting from a server-based architecture to a serverless model to speed-up the pace of technology transformations. A recent Cloud Foundry global survey of 600 IT decision makers found that 19% of respondents are already using serverless technologies and that the number is expected to increase by 42% in the next two years[1].

Demand for serverless technologies is on the rise because it provides the opportunity for faster time-to-market, by dynamically and automatically allocating compute and memory based on user requests. It also provides cost savings through hands-off infrastructure management, which enables companies to redirect IT budget and human capital from operations to innovation. The pay-as-you go model with serverless technologies leads to a shift from large capital expenditure lock up to flexible on-demand consumption, allowing users to scale, customize, and provision computing resources dynamically to meet their exact needs. This, in turn, impacts business agility.

However, it can be difficult to estimate costs in a serverless model because inputs are variable. In this white paper, we will introduce a framework for comparing the total cost of ownership for both serverless and traditional applications, factoring in infrastructure, development, and maintenance costs. We evaluate the financial impact and business value of both a traditional server-based architecture (with Amazon EC2 instances) and a serverless model (with AWS Lambda functions). Based on this model and the applications evaluated, we see that while infrastructure costs may be higher with a serverless approach, the total cost of ownership is significantly lower with serverless due to savings in development and maintenance costs.

Introduction of the Serverless TCO Framework

Serverless technologies effectively shift operational responsibilities to a cloud service provider, and companies are applying this philosophy across the entire application stack, including compute, storage, and network. With a serverless operational model, there are no servers to provision, patch, or manage and there is no software to install, maintain, or operate. In summary, a serverless model enables enhanced scalability, agility, and resiliency and allows developers to instead have a greater focus on core value-added tasks. Many organizations who take advantage of serverless technologies can deploy more frequent releases of their products and services, thereby impacting faster time-to-market and accelerated revenue growth.

Based on our extensive experience working with Fortune 100 clients across industries, we have developed a Serverless Total Cost of Ownership (TCO) framework to evaluate the true cost of running a net new application using serverless technologies, such as AWS Lambda or Kinesis in comparison to a traditional compute, such as Amazon EC2. The Serverless TCO framework is comprised of three key cost components: infrastructure, development, and maintenance.

[1] Deloitte Whitepaper on Technology Trends 2019

1. **Infrastructure cost** is charge incurred from hosting an application workload on a cloud service provider, in this case, Amazon Web Services (AWS)

   The detailed section further in this paper emphasizes two Deloitte client examples:

   a. *Comparing AWS Lambda functions vs Amazon EC2 instances for a transportation client*

   b. *Comparing Amazon Kinesis vs Hadoop Clusters on EC2 for a global banking client*

2. **Development cost** is the upfront charge of building and developing a new application on a cloud-based service

   The detailed section in this paper highlights Deloitte's industry experience estimating development time and the cost of an average development resource

3. **Maintenance cost** is the day-to-day operations expense associated with running and maintaining an application on an EC2 instance vs. serverless architecture

   This section of the paper shows typical Deloitte benchmarks for maintenance costs across various components, including traditional security, patching, service ticket, and testing teams

While there are organizational benefits of moving to serverless, such as increased velocity to address business opportunities, better planning of infrastructure capacity, etc., this paper focuses only on the cost elements highlighted above.

Infrastructure

This is the first major cost component and is comprised of the compute, storage and network services consumed to host application workloads on the AWS cloud platform. Infrastructure costs are often referred to as the "Cost to Run" the application workloads.

- Compute costs in an Amazon EC2 environment are calculated based on the maximum number of requests an instance can process per second, the number of servers to accommodate peak traffic (web, apps, database), and the time-period an instance is active.
- In a serverless model, infrastructure cost is based on actual execution time, i.e. the application owner is only charged when the code is executed effectively achieving a 100% server utilization (AWS Lambda, for example, is charged based on number of requests and duration of the request).
- In addition, leading practices such as high availability/fault-tolerance, load balancing, and security services are included in the serverless architecture whereas those services would require additional charges in the traditional cloud environment.

Note - Common services, such as API gateway costs, data transfer, storage, database, and other cloud services pricing that are used by both EC2 and serverless applications, are explicitly left out of the calculations as they apply equally to both on-premise and public cloud environment.

To analyze the Infrastructure costs, we used two real-life client examples:

Case Study 1: Transportation Company evaluating AWS Lambda over Amazon EC2

Overview

The average commuter for this transportation client spends about two hours in transit, during which they can book tickets online, connect to wifi, and can monitor their trip in real time. Now multiply that by millions of passengers annually, hundreds of destinations, and thousands of routes. Transportation companies supporting these passengers typically struggle with legacy systems that are expensive to support and update. This can lead to increasingly unpredictable and slow response times which cause reports to be delayed and obsolete. These companies are increasingly moving to a serverless model to reduce the burden of infrastructure management. This is made possible by utilizing a host of microservices that only execute when needed, which produces the required data rapidly and seamlessly.

For the purposes of this paper, we compared the costs incurred by the client as they evaluated whether to run their ticket booking system through Lambda functions or on traditional EC2 instances.

Cost Calculations

The transportation company chose AWS Lambda, among other serverless components, to process bookings and tickets for all its users. With about 1.5M transactions in a day, this application consumed about $1090 per month in combined infrastructure cost between two lambda functions as shown in the table below. Due to the architectural requirements and decoupled microservices best practice guidance, two separate lambda functions were needed—one for ticket processing for the downstream customer, and the other for data processing (analytics and reporting).

If the same application were deployed to run in traditional server-based infrastructure, we have assumed the need for three m5 large EC2s as webservers (in lieu of the lambda function for downstream customer) and three r5 large EC2s as database servers along with some dedicated EBS storage for fast data access (in lieu of the lambda function for data analytics and reporting). The cost to run this application amounted to $790.

Monthly Compute costs for EC2 vs Lambda can be compared as follows:

Compute Costs	Traditional Cloud (EC2)	Serverless (Lambda)
Webservers (3 x m5 large)	$210	NA
Database Server (3 x r5 large)	$276	NA
1GB 500 IOPS Provisioned IOPS SSD	$33	NA
Load Balancing (for availability between three servers)	$246	NA
AWS Lambda Function 1 (Memory Allocation: 512 MB); Execution time (3000 ms) for ticket processing for downstream customer	NA	$720
AWS Lambda Function 2 (Memory Allocation: 512 MB); Execution time (3000 ms) for data processing	NA	$270
Total Monthly Cost	**$790**	**$1090**
	Monthly Cost Difference	**$300**

Assumptions

Monthly requests/events:	43,200,000
Peak to average ratio:	3:1
Avg. request size (KB):	2
Avg. response size (KB):	20

Case Study 2: Banking Company evaluating AWS Kinesis vs Hadoop Clusters on EC2

Overview

On any given business day, millions of trades are made for billions of shares on any one of the stock exchanges. Financial institutions making and receiving trades require real-time data analytics capabilities to process the thousands of data sets per second. The challenge for IT in these organizations is to provide seemingly limitless capacity for dynamic search capabilities. Those with on-premises or server-based environments lose the ability to auto-scale and perform real-time analytics to proactively check for anomalies or fraud with data sets. Serverless technologies provide automated scaling in support of search capabilities, above and beyond a basic query, enabling companies to react in real-time.

For the purposes of this paper, we compare the costs incurred by the client as they evaluated whether to run data queries through Apache Hadoop Clusters on EC2 instances or through AWS Kinesis.

Cost Calculations

The banking company employed Kinesis, AWS's fully managed data-query service to perform real-time data streaming and analysis. Based on the assumptions that the client needs to run about 500 queries a day (5 GB of data scanned per query), the infrastructure cost of running Kinesis was about $380 per month. For the same application deployed to run in traditional EC2, we have assumed the need for three r5 data servers running Apache Hadoop. The table below a comparative analysis between the two platforms for this scenario.

Monthly Compute costs for Hadoop on EC2 vs AWS Kinesis can be compared as follows:

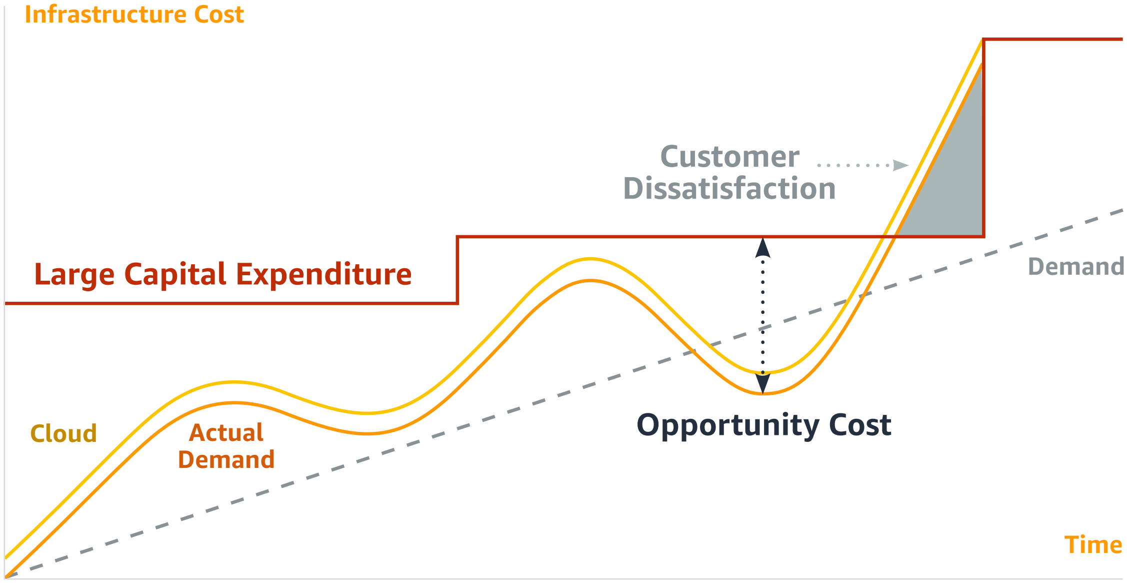
Compute Costs	Traditional Cloud (EC2)	Serverless (Kinesis)
Database Server (3 x r5 large)	$276	NA
16GB General Purpose SSD	$2	NA
Load Balancing (for availability between three servers)	$18	NA
Kinesis (Monthly Data Scanned: 75TB); Cost per TB scanned ($5)	NA	$378
Total Monthly Cost	**$296**	**$378**
	Monthly Cost Difference	**$82**

In both use cases, when we look at just the infrastructure cost, it was more economical to implement this application in a traditional EC2 environment. However, to understand the total cost of running an application as server-based vs. serverless, we need to compare not just the infrastructure costs, but also the development and maintenance cost in their respective sections as shown below.

Development

The second major cost component, development, is an upfront one-time cost that can be quantified by the time and effort required for pre-planning the application build. This is often referred to as the "Cost to Achieve" migration to Cloud. Using EC2 instances, one must determine how the architecture should scale to support the application over time. However, in a serverless environment, the capacity is auto-scaled so that fluctuations in demand are accommodated. As expected, if we under scale an EC2 instance we will be faced with challenges surrounding our ability to provide enough capacity when we experience peak activity. However, if we overscale an EC2 instance we will be spending more money than required as we have underutilized capacity.

The diagram below highlights the benefit of serverless in dynamically auto-scaling to track the actual usage requirements.



An EC2 environment secures some fixed capacity based on projected forecasts and does not scale as dynamically as serverless applications. This can lead to waste in spend when capacity is over provisioned (opportunity costs) and customer dissatisfaction when capacity does not meet usage requirements.

- Developers using EC2 instances need to spend considerable time evaluating challenges that the IT architecture could face at scale and determine what tradeoffs need to be made upfront
  - The cost incurred for this preplanning includes the number of resources involved and the cost of both resources and time
- Additional costs are incurred due to developer time spent setting up network and load balancers, provisioning auto-scaling, planning for availability (selecting the right number of Availability Zones), and purchasing the licenses and software

A serverless application leverages an event-based architecture, thereby allowing development teams to start developing the application rather than planning a robust deployment architecture. The table below summarizes the typical savings we have seen from the reduction of time required to provision applications on Serverless vs traditional EC2 instances. On average, a serverless environment takes 68% less time to provision as compared to an instance-based environment, which can equate to hundreds of dollars in savings per month per application[2].

One-Time Development cost for Traditional (EC2) compared with Serverless (AWS Lambda)

Development	Traditional Cloud	Serverless	Difference
Days to Deploy	~25 days	~8 days	~17 days
Up Front One-Time Cost	$38,300	$12,300	$26,000
Monthly Cost	$640	$205	**$(435)**

Monthly Cost calculated from Annual FTE Cost per month amortized over 5-years

- Days to deploy new compute/storage
- Traditional: 3 developers took 4-5 weeks
- Serverless: 3 developers took 8-9 days
- FTE Rate $120K/yr., 8 hours/workday
- One-Time Fee is for 3 developers, 8-hour days
- Monthly cost is assumed to amortized over 5-years
- Net new application - no application migration costs
- Assuming the right talent exitsts - thereby no additional cost to hire/train developers
- Building a stateless application
- Not a consistently high memory usage application
- Not a consistently high CPU application
- Not a near real-time application – e.g. running in stock exchanges

[2] IDC: Generating Value Through IT Agility and Business Scalability with AWS Serverless Platform

Maintenance

The third major cost component, maintenance, takes into account the time and resources spent on ongoing tasks once the application is deployed in production, which is commonly referred to as the "Cost to Support" an application. Maintenance cost can be categorized as time spent by a developer in 4 areas:

1. Provisioning and scaling of applications
2. Security implementation (hardening of AMIs)
3. Patching and Operating system updates
4. On-going application operations, such as delivering/adding new features, monitoring, logging, verifying and testing

Although numbers vary based on the client organization and nature of the application, on average, Deloitte estimates that an application developer spends about 8-10 hours a month on application provisioning, security implementation, and patching and OS updates. An additional 40 hours a month is spent in application monitoring, logging, verifying, and testing when running EC2 services.

With serverless capabilities, such as Lambda and Kinesis, most of these maintenance tasks are no longer required because these services are fully managed by the cloud provider. This allows the developers to focus their time and resources on developing the core capability to create or build their business, verses focusing on reboots and reconfigurations on the servers themselves.

- In a traditional EC2 model, teams would need to open service tickets and patching teams would reach out to developers to patch the environment, all of which could then delay development activities

- In the serverless model utilizing Lambda, these types of patches and other related activities happen behind the scenes to not impact core development

Additionally, serverless implementations can digitize many security rules, thus making them more secure and eliminating the need for human intervention and resource requirements, such as housing a dedicated team to handle special provisioning of firewall licenses and host scanning. The table below summarizes the additional time (in hours per month) of application maintenance it may take an app developer.

Ongoing Maintenance efforts for On-Prem vs Traditional (EC2) vs Serverless (AWS Lambda) for an entire application portfolio

Maintenance Costs	Traditional Cloud (Hours)	Serverless (Hours)
Provisioning & Scaling	8	1
Security Implementation	8	1
Patching & OS updates	8	1
On-going application operations	40	8 - 32
Monthly Development Cost in App Maintenance	$4096	$704 - $2240
Monthly Cost Difference		**$(3,392) - $(1,856)**
% savings (moving to serverless from EC2)		**45% - 80%**

Assumptions

Monthly Cost calculated from Annual FTE Cost per month

- FTE Rate $120K/yr $64/hr

Conclusion

In the above sections, we compare the TCO framework across two Deloitte client use cases, to demonstrate how we evaluate the total cost of a net new application on EC2 instances vs. on Serverless services. When considering only the infrastructure cost, running the application on EC2 instances is the more cost-effective choice. However, when we account for development and maintenance costs, it becomes significantly cheaper to run the application through serverless technologies, such as, Lambda or Kinesis. In both use cases, the client decided to build with serverless architectures to realize these cost savings. The table below summarizes the total combined costs across both uses cases:

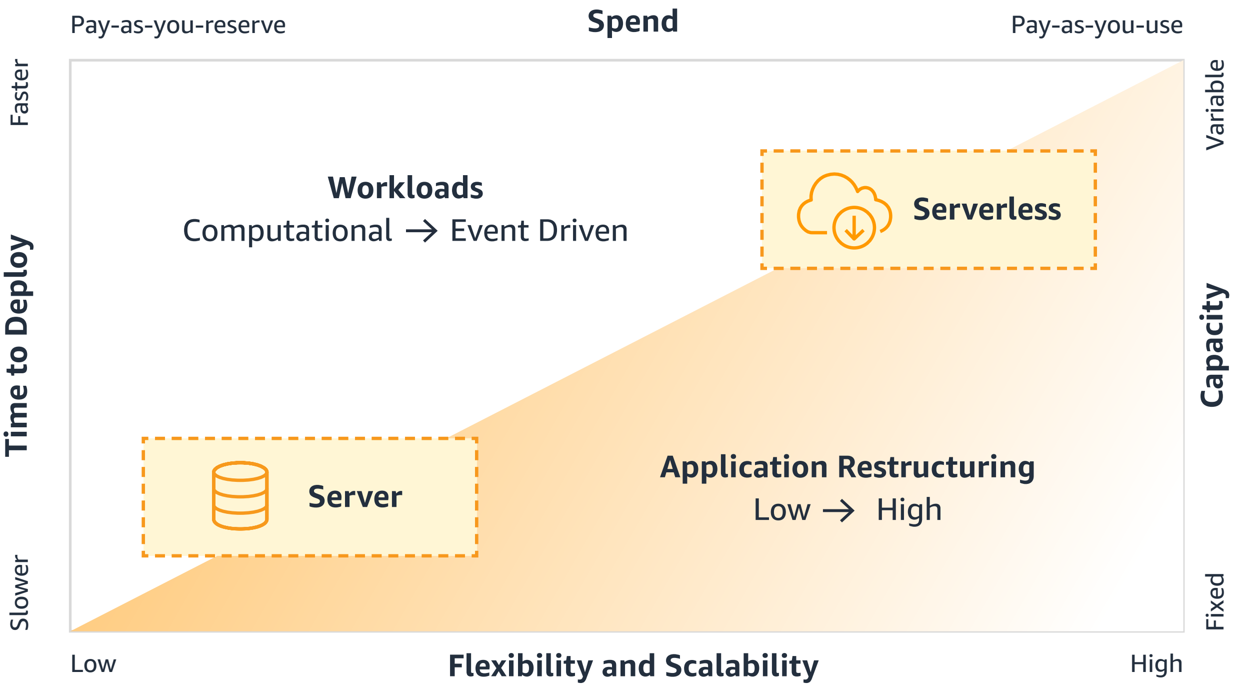
	Transportation Company		Banking Company	
	EC2	**Serverless**	**EC2**	**Serverless**
Infrastructure Cost ($/month)	$790	$1090	$296	$378
	Difference	**$300**	Difference	**$82**
Development Cost ($/month)	$640	$205	$640	$205
	Difference	$(435)	Difference	$(435)
Maintenance Cost ($/month)	$4096	$2240	$4096	$2240
	Difference	$(1856)	Difference	$(1856)
Total Cost ($/month)	$5526	$3535	$5032	$2823
	Difference	$(1991)	Difference	$(2209)

Therefore, when considering the cost of running an application in a traditional cloud environment like EC2 over serverless architectures such as Lambda or Kinesis, it is important to consider the total cost of running the application which includes the infrastructure, development, and maintenance costs, also referred to as the cost to run, cost to achieve, and cost to support the application. In isolation, each one of these cost components may provide an incomplete picture of the total cost and a comprehensive comparison across all three cost components is needed to arrive at an accurate total cost of ownership (TCO).

Exceptions to the rule

Although there are many benefits of moving to serverless technologies, not all applications are the right fit for a serverless architecture. It is important to consciously select your technology stack and configure it in a cost-effective manner for serverless functions to yield the fully intended benefits. As the diagram below illustrates:

- Applications with variable capacity and high scalability requirements are good candidates for serverless
- Applications that spend large amounts of time running, while waiting on an event to be triggered, will continue paying for their API and other service calls, thus making them not ideal candidates for serverless
- Serverless is best suited for web, mobile, and IoT apps, real-time analytics, and data processing
- Serverless may be least suited for long-running or complex computational tasks, data migrations from relational to NoSQL, applications requiring significant disc space or RAM, and applications requiring SSH server access.



In summary, time spent on maintenance and ongoing operations is diminished in serverless applications, as this is fully managed by the cloud provider. As such, the roles of a dedicated operations team will need to evolve. Serverless architectures also provide infinite scalability and built-in high availability, which are additional efforts and costs in a traditional environment. When we compare only infrastructure costs across the platforms we may determine that a traditional model is cost-effective. However, when we layer on the additional benefits and cost savings of a serverless model, organizations can save significantly by constructing applications and overall organizational structures to effectively leverage a serverless architecture.

Produced in partnership with:

aws

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

About Deloitte
Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. In the United States, Deloitte refers to one or more of the US member firms of DTTL, their related entities that operate using the "Deloitte" name in the United States and their respective affiliates. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see www.deloitte.com/about to learn more about our global network of member firms.

Deloitte.