

# Serverless Computing: The Future of Software Architecture

Néstor Raúl Valdez Salinas

Facultad de Ciencias y Tecnología, Universidad Católica Nuestra Señora de la Asunción

## 1 Introducción

La optimización de costos, una mejor flexibilidad y una gestión eficiente de los recursos son algunos de los factores clave que alimentan el crecimiento de la infraestructura de la nube. La informática Serverless es otro paso en esta dirección[1].

Serverless Computing, ha aumentado dramáticamente en la popularidad desde la introducción de la plataforma de Lambda por AWS en 2014. De allí en mas varios otros proveedores se fueron sumando. El nombre "sin servidor" claramente no sugiere que no existan servidores para los cómputos, sino que estos son transparentes para el consumidor del servicio. No hay máquinas virtuales expuestas. Esto significa: no aprovisionamiento de Infraestructura ni gestión continua de los recursos computacionales. La ejecución de las funciones de los usuarios se abstraen de las máquinas en las que se realiza, y como tal se puede ver para ofrecer una plataforma de ejecución "pura". Aunque muchos proveedores de servicios en la nube ahora ofrecen plataformas de ejecución sin servidor, este documento se centra en AWS lambda, Microsoft Azure e IBM Bluemix como los principales proveedores y además lo que tienen una mejor Proyección para el futuro[2]. Se hablara de los servicios que cada uno ofrece además de las ventajas y deventajas que implica el uso de esta arquitectura.

## 2 Serverless Computing

### 2.1 Origen de "Serverless"

El termino "Serverless" es confuso ya que debe existir tanto servidores (hardware) como tambien procesos corriendo en alguna parte. Lo que diferencia a "Serverless" de un uso mas tradicional es que el(los) desarrollador(es) no estan pendiente(s) a el hardware o los procesos que corren en estos servidores, esto queda relegado al proveedor del servicio.

Los primeros usos del termino aparecieron en el año 2012, en un articulo de Ken Fromm<sup>1</sup>. Empezamos a ver este termino siendo usado con mas frecuencia en 2015, luego que Amazon lanza AWS Lambda en el año 2014. En el año 2015 hubo una charla en Amazon titulada "The Serverless Company using AWS Lambda". Hacia el final del año 2015 el proyecto Open Source JAWS<sup>2</sup> fue renombrado Serverless Framework, continuando la tendencia.

En el año 2016 se ven varios ejemplos, como la Serverless Conference, ademas de varios proveedores que adoptaron el termino para describir a sus productos [3]

### 2.2 Qué es Serverless?

"Serverless" fue usado por primera vez para describir a aplicaciones que dependian significativamente o totalmente de un servicio tercerizado ("en la nube") para manejar toda la logica del lado del servidor. Nos referimos a servicios como acceso a base de datos en la nube y servicios de autenticacion. Estos tipos de servicios se demoninan Backend as a Service o 'Baas'.

Serverless tambien puede referirse a aplicaciones donde una parte de la logica del lado del servidor es aun manejada por el desarrollador pero a diferencia de una arquitectura tradicional corren en contenedores que son event-triggered, y manejadas totalmente por un tercero. Esto se demonina 'Function as a service' o 'FaaS'. AWS Lambda es uno de los proveedores mas populares que ofrecen este tipo de servicio[3].

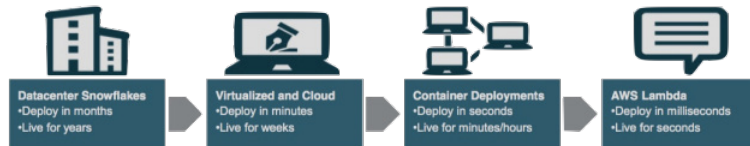
Entonces podemos decir que Serverless Computing es una arquitectura dónde la ejecución del código está totalmente manejada por un proveedor en la nube, en lugar de la forma tradicional de desarrollar aplicaciones y montarlas en servidores uno mismo. Esto implica que toda la tarea de mantenimiento del servidor es totalmente abstracto para el desarrollador [4].

---

<sup>1</sup> Why The Future of Software and Apps is Serverless. <http://readwrite.com/>

<sup>2</sup> Javascript Amazon Web Services

Fig. 1. Evolucion del desarrollo



### 3 Proveedores de Serverless Computing

Actualmente varias compañías ofrecen el servicio de Serverless Computing, entre ellas quizá las más populares de todas es AWS Lambda de Amazon. Microsoft e IBM también ofrecen servicios parecidos con Azure y Blumix respectivamente.

#### 3.1 Amazon Web Services Lambda

AWS Lambda es un servicio de cómputo en el que puede cargar código a AWS y el servicio se encarga de la ejecución del código en la infraestructura de AWS. Después de cargar el código y crear lo que llamamos una función Lambda, AWS Lambda se encarga de aprovisionar y administrar los servidores que se utiliza para ejecutar el código.

Permite ejecutar el código en una infraestructura de cómputo de alta disponibilidad y realiza toda la administración de los recursos computacionales, incluidos el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalamiento automático, la supervisión de código y registro. Todo lo que el desarrollador necesita hacer es suministrar el código a AWS Lambda en alguno de los lenguajes que este maneja (actualmente `node.js`, Java y Python).

AWS ejecuta su código sólo cuando es necesario y escala automáticamente, de unas cuantas solicitudes por día a miles por segundo. Con estas capacidades, puede utilizar lambda para crear fácilmente desencadenadores de procesamiento de datos para servicios AWS como Amazon S3 y Amazon DynamoDB, procesar los datos de streaming almacenados en la cinética de Amazon, o crear su propio back-end que opera a escala de AWS, performance y seguridad[6]. AWS cobra

por la potencia de computo que se utiliza, en incrementos de 100 milisegundos. Los desarrolladores pueden concentrarse en su código y desencadenadores de eventos y AWS se encarga del resto[4].

### 3.2 Microsoft Azure

Microsoft Azure, anteriormente conocida como Windows Azure, es la plataforma de Cloud Computing de Microsoft. Proporciona una gama de servicios en nube, incluyendo computo, analytics, almacenaje y la red. Los usuarios pueden elegir entre estos servicios para desarrollar y escalar nuevas aplicaciones, o ejecutar aplicaciones existentes, en la nube pública. Microsoft introdujo Azure en octubre de 2008. La plataforma de nube se llamaba originalmente Windows Azure, pero fue remarcado a Microsoft Azure en abril de 2014. Para garantizar la disponibilidad, Microsoft tiene centros de datos Azure ubicados en todo el mundo. A partir de enero de 2016, Microsoft dijo Azure Services están disponibles en 22 regiones de todo el mundo, incluyendo en los Estados Unidos, Europa, Asia, Australia y Brasil.[7].

Microsoft Azure es visto como una Platform As a Service o 'PaaS' y como Infrastructure as a Service o 'IaaS'. Los principales servicios que Azure ofrece son[7][8]:

- **Computo:** provee de maquinas virtuales, contenedores, procesamiento por lotes y acceso remoto a aplicaciones.
- **Web y Mobile:** estos servicios apoyan el desarrollo y despliegue de aplicaciones web y móviles, y también ofrecen funcionalidades para la gestión de API, notificación y reportes.
- **Data Storage:** incluye base de datos como un servicio (SQL y NoSQL) así como almacenamiento en la nube no estructurado.
- **Analytics:** estos servicios proporcionan análisis y almacenamiento de información distribuidos, así como análisis en tiempo real, análisis de datos grandes, lagos de datos, aprendizaje de máquinas y almacenaje de datos.
- **Development:** este servicio permite a los desarrolladores compartir código, probar aplicaciones y rastrear posibles problemas. Azure soporta una amplia gama de lenguajes de programación de aplicaciones, incluyendo JavaScript, Python, .net y node. js.

### 3.3 IBM Cloud Bluemix

IBM Bluemix es una plataforma en la nube que ayuda a los desarrolladores a crear y ejecutar aplicaciones y servicios modernos. Bluemix proporciona a los desarrolladores acceso instantáneo a los sistemas y servicios que necesitan para ejecutar rápidamente, iterar continuamente y escalar correctamente. Con sus servicios en dispositivos móviles, Internet de las cosas, Watson y más, IBM Bluemix es la plataforma ideal para impulsar las innovadoras aplicaciones basadas en datos[9]. Bluemix combina PaaS<sup>1</sup> e IaaS<sup>2</sup>.

<sup>1</sup> Platform as a Service

<sup>2</sup> Infrastructure as a Service

IBM permite:

- Implementar la infraestructura de almacenamiento de información y cómputo de alto rendimiento en los centros de datos de Cloud de IBM seguros en todo el mundo.
- Probar y adoptar una amplia gama de servicios y capacidades en la nube de IBM, comunidades de código abierto y desarrollo de terceros.
- Conectar todos los sistemas y aplicaciones desde una plataforma única, escalable y en la nube.

## 4 Ventajas y Desventajas frente a los métodos tradicionales

### 4.1 Ventajas

El optar por utilizar alguno de los proveedores de Serverless Computing tiene ventajas y desventajas en distintas áreas, como puede ser en costo, operación y productividad.

**Costo** Serverless Computing es más eficiente en la relación costo-eficiencia, de lo que sería rentar o comprar una serie de servidores/procesadores. En las arquitecturas de servidores tradicionales, el tráfico de ráfagas significa que se debe crear el servidor para controlar las tasas de ráfagas máximas. Pero el resto del tiempo, estás desperdiciando dinero con ciclos de CPU inactivos. En lugar de tener que pagar por esa ociosidad, una arquitectura sin servidor le permite sólo pagar por lo que los ciclos de CPU que realmente consume y el código sólo se ejecuta cuando sea necesario[11].

**Operación** A pesar de que la virtualización de servidores, el sistema operativo, el tiempo de ejecución, las bases de datos y otros componentes de infraestructura siguen siendo técnicamente requeridos, una arquitectura sin servidor elimina la administración de la pila de servidores y cualquier inquietud/planificación que tenga que ir a la escalabilidad potencial hacia arriba o hacia abajo de la pila. Todos estos procesos son automatizados y usted simplemente paga por el tiempo de cálculo que utiliza[11].

**Productividad** La tarea de los desarrolladores de back-end se ve disminuida puesto que no tienen que preocuparse de por ejemplo la programación multihilos o manejar peticiones HTTP. Esto es debido a que en Serverless Computing las unidades de código expuestas al mundo exterior son funciones sencillas.

Otras ventajas incluyen que dos de los proveedores más famosos con Google Cloud y Amazon Lambda ofrecen soporte para varios lenguajes de programación. En el caso de Google es JavaScript y AWS Lambda soporta JavaScript, Java, Python y JRuby[11].

## 4.2 Desventajas

El optar por una arquitectura Serverless en una u otra medida también tiene sus desventajas en performance, seguridad entre otras.

**Performance** El modelo en sí mismo significa que se obtendrá una mayor latencia en la forma en que los recursos de cálculo responden a los requisitos de las aplicaciones. Si el rendimiento es un requerimiento, es mejor utilizar servidores virtuales asignados[10].

**Monitoring and Debuging** El monitoreo y la depuración también es complicado. El hecho de que no esté utilizando un único recurso de servidor hace que ambas actividades sean muy difíciles[10].

**Limitacion de lenguajes.** A pesar de la capacidad de programar con muchos de los lenguajes más populares, Serverless computing sigue siendo limitado, considerando todos los lenguajes disponibles en los que se puede codificar, compilar y hacer funcionar en ambientes tradicionales. Además de eso, sólo hay unos pocos proveedores respetables de servicios de Serverless Computing[11].

**Seguridad.** Serverless Computing a veces se considera, erróneamente, como más seguro que las arquitecturas tradicionales. Aunque esto es cierto hasta cierto punto, porque las vulnerabilidades del sistema operativo son atendidas por el proveedor de la nube, la superficie de ataque total es significativamente mayor ya que hay muchos más componentes en la aplicación en comparación con las arquitecturas tradicionales y cada componente es un punto de entrada a la aplicación Serverless. Además, las soluciones de seguridad que los clientes utilizan para proteger sus cargas de trabajo, en la nube se vuelven irrelevantes ya que los clientes no pueden controlar e instalar nada en el punto final y el nivel de red como IDS/IPS.

## 5 Conclusion

Los servicios "Serverless" han demostrado ser de mucha utilidad desde que AWS lo introdujo con fuerza en el año 2014. Ese fue el puntapie inicial para que otras empresas comiencen a invertir en infraestructura para poder ofrecer servicios a la misma altura. Las ventajas y facilidades que ofrece a los desarrolladores estan mas que claras, pese a que, como cualquier tecnologia, cuenta con desventajas, la cantidad de beneficios desde el costo de operaciones y las abstracciones que los desarrolladores pueden tener superan, desde mi punto de vista, a las desventajas o limitaciones que se puedan presentar. No hay dudas que cada vez mas desarrolladores empezaran a implementar los servicios de Serverless Computing para el despliegue de cualquier aplicacion, movil o web.

## References

1. Shiv Singh. (April 23rd, 2017) Serverless Computing. <http://blogs.perficient.com>
2. Contino. Introduction to Serverless Computing with AWS Lambda. <https://www.contino.io/>
3. Mike Roberts. (August 04, 2016) Serverless Architectures. New York. USA. <https://martinfowler.com>
4. Scott Carey. (Julio 10, 2007) What is serverless computing? Here's how AWS Lambda is helping enterprises run it. UK. <https://www.computerworlduk.com>
5. Amazon Web Services. AWS Lambda. <http://docs.aws.amazon.com>
6. Abhinary Reddy. (January 20, 2016) What exactly is Amazon Lambda? How does it work? <https://www.quora.com>
7. Margaret Roques. (February 2016) Microsoft Azure (Windows Azure). <http://searchcloudcomputing.techtarget.com>
8. Microsoft Company. (2017) Microsoft Azure. <https://azure.microsoft.com>
9. IBM. (2017) IBM Bluemix. <https://www.ibm.com>
10. David Linthicum. (January 6, 2016) Cloud Computing: The Pros and Cons of Serverless Computing. InfoWorld. <https://www.infoworld.com/>
11. Andrew Froehlich. (August 22, 2016) Serverless Architecture Pros and Cons. <http://networkcomputing.com>