



APACHE OpenWhisk™

Una tecnología Serverless.

Luis Manuel Raya Bernal
IES Gonzalo Nazareno

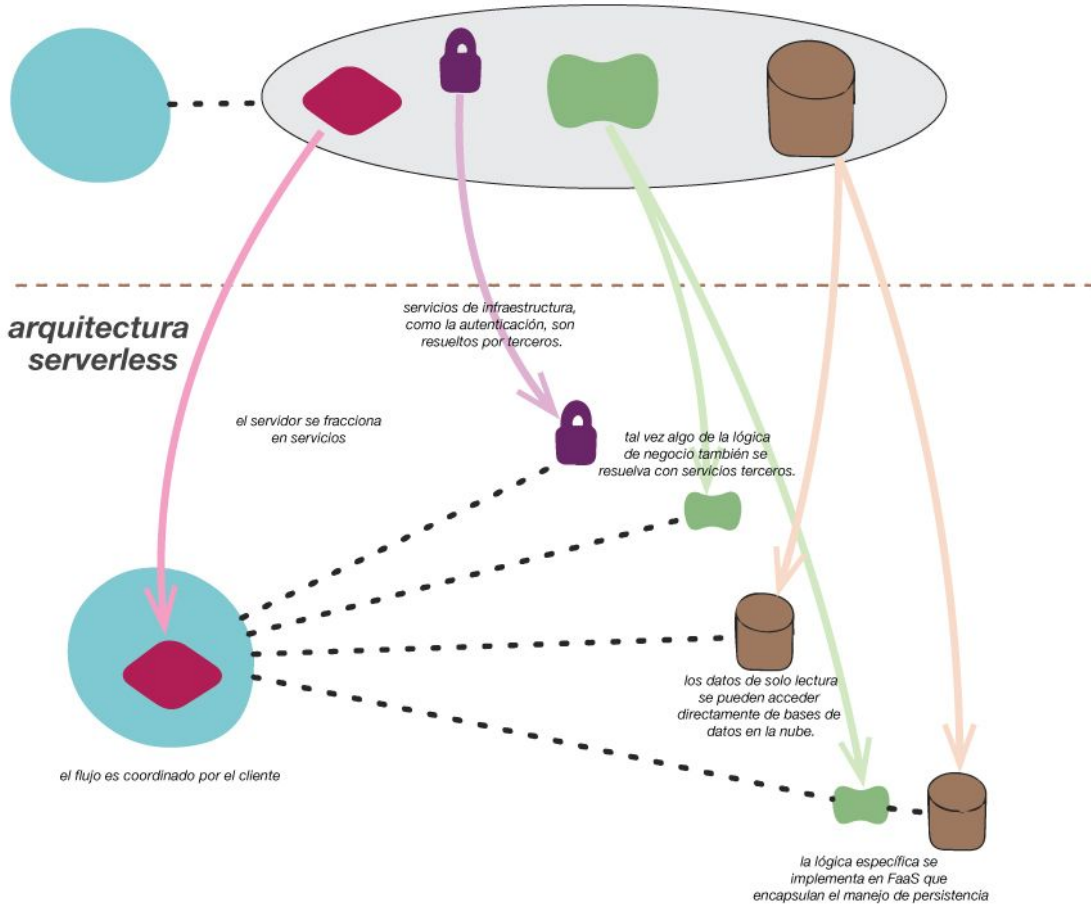


¿Que es Serverless?

Aunque el nombre es contradictorio, en la arquitectura Serverless si hay servidores, la diferencia es que nosotros no tenemos el control, solo podemos hospedar servicios y nuestro código.

En estos servidores no tenemos que gestionar ni administrar ningún servidor y lo mejor de todo, podemos escalar manual o automáticamente adaptándose a la demanda.

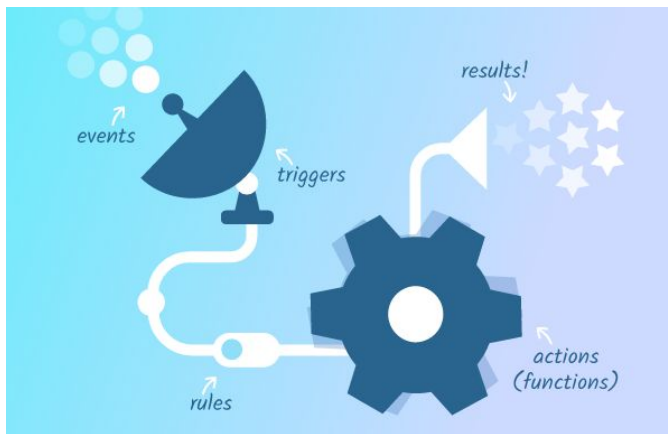
En una aplicación tradicional el cliente se comunica con un proceso en el servidor que se encarga de coordinar todos los aspectos de lógica de la aplicación.



¿Que es FaaS?

Se asocia **Serverless** con **FaaS (Function as a service)** que es una tecnología que nació en **2014** en **hook.io**

El mercado para reducir costes se ha centrado en **la función**, por esto el servidor se encarga de codificar la función y un sistema externo se encarga de ejecutarla en el momento adecuado, esto es FaaS





Empresas que disponen de Serverless





¿Ventajas? ¿Inconvenientes?

El principal beneficio de esta arquitectura es el ahorro de costes, este modelo de negocio **solo cobra por peticiones atendidas** y el **uso de espacio en disco**, además de esto se ahorra costes de **personal y administración** de servidores.

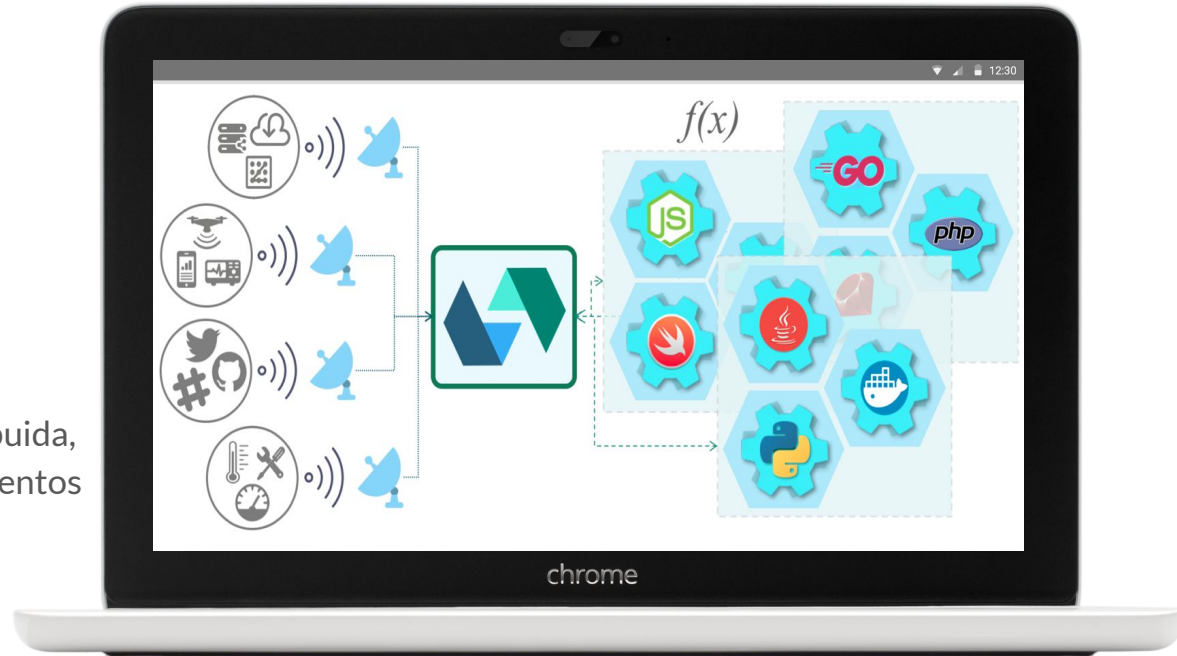
- Ninguna configuración o mantenimiento.
- Escalabilidad.
- Cobro por consumo de forma lineal, si no se ejecuta no se paga.
- Entorno limitado al lenguaje y todo lo que tenga el proveedor.
- Servicio sin estado, la persistencia de información la aportan otros servicios.
- Se cobra por tiempo de ejecución, algunos proveedores limitan el tiempo, los proveedores que hacen esto penalizan los consumos prolongados en el tiempo.

Apache OpenWhisk

Una plataforma FaaS, OpenSource y distribuida, que ejecuta las funciones en respuesta a eventos de cualquier escalado.

Usa el modelo de programación:

T(trigger) R(rules) A(actions) en cualquier lenguaje.



Ejemplo: Un widget del tiempo en nuestro móvil, usaría el trigger para detectar la ubicación y sus cambios.

Tras esto se desarrolla la acción que llamaría al servicio del tiempo y obtendría la previsión del tiempo. Y para acabar se crearía la regla que asocia al trigger y la acción.



¿Cómo funciona?

Vamos a usar un Hello World de un cluster local para ver como funciona internamente:

```
def main(args):  
    name = args.get("name", "stranger")  
    greeting = "Hello " + name + "!"  
    print(greeting)  
    return {"greeting": greeting}
```

Creación de la acción: `wsk action create helloPython hello.py -i`



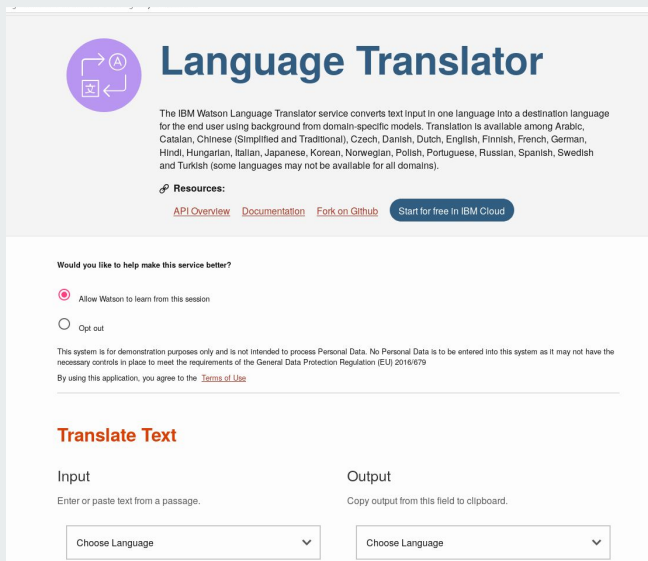
¿Cómo funciona?

Llamaremos a la acción y nos dará un JSON:

```
wsk action invoke --result helloPython --param name World -i
```

```
{  
  "greeting": "Hello World!"  
}
```

Demostración de una app 100% Serverless



The screenshot shows the IBM Watson Language Translator web interface. At the top left is a purple circular icon with a white 'A' and a double-headed arrow. To its right is the title 'Language Translator' in blue. Below the title is a paragraph of text explaining the service. Underneath is a 'Resources' section with links for 'API Overview', 'Documentation', and 'Fork on Github', and a 'Start for free in IBM Cloud' button. A survey question asks 'Would you like to help make this service better?' with two radio button options: 'Allow Watson to learn from this session' (selected) and 'Opt out'. Below the survey is a disclaimer about personal data. The main section is titled 'Translate Text' and contains two columns: 'Input' and 'Output'. Each column has a text area and a 'Choose Language' dropdown menu.

Language Translator

The IBM Watson Language Translator service converts text input in one language into a destination language for the end user using background from domain-specific models. Translation is available among Arabic, Catalan, Chinese (Simplified and Traditional), Czech, Danish, Dutch, English, Finnish, French, German, Hindi, Hungarian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Russian, Spanish, Swedish and Turkish (some languages may not be available for all domains).

Resources:

[API Overview](#) [Documentation](#) [Fork on Github](#) [Start for free in IBM Cloud](#)

Would you like to help make this service better?

Allow Watson to learn from this session

Opt out

This system is for demonstration purposes only and is not intended to process Personal Data. No Personal Data is to be entered into this system as it may not have the necessary controls in place to meet the requirements of the General Data Protection Regulation (EU) 2016/679. By using this application, you agree to the [Terms of Use](#).

Translate Text

Input
Enter or paste text from a passage.

Output
Copy output from this field to clipboard.

Choose Language

Choose Language

En esta demostración del uso de un servidor Serverless, voy a usar la nube de IBM, ellos tienen implantado OpenWhisk de forma comercial

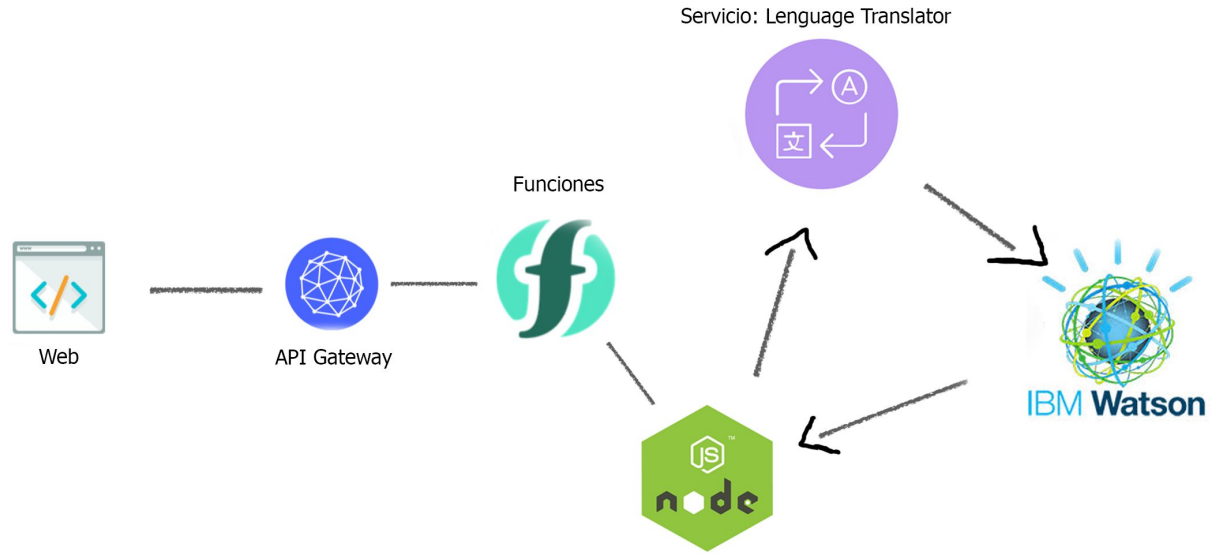
Para esta prueba he necesitado:

- Una instancia de Apache OpenWhisk en IBM Cloud
- IBM Cloud CLI
- Crear una instancia de Watson Language Translator

¿Quieres probarlo? Accede:

<https://language-translator-serverlessow.eu-gb.mycluemix.net>

Arquitectura de la app





Turno de preguntas

Gracias por descubrir este nuevo paradigma.

Has pensado que si el Serverless no necesita gente de sistemas...
Tenemos que renovarnos como DevOps?