Serverless Architectures: The (r)evolution of Cloud Resource Management

Martin Garriga

Assistant Researcher – CONICET

Lecturer – Faculty of Informatics

2018 Devfest GDG – Facultad de Informática

About me

- Professor and Researcher in National University of Comahue, and CONICET
- Specialized in traditional SOA (Service-Oriented Architectures)
- From 2016 to 2018, Posdoc@Polimi, working on **Microservices and Serverless Architectures**

Roadmap

- Serverless Architectures Definition
  - Serverless, FaaS, Lambda Functions??
  - Evolution of Cloud Platforms
- Serverless Features
  - Zero-management
  - Self-scaling and provisioning
  - Cost Model
  - Performance
  - Availability
- Let’s Practice! Functions@FIREBASE
- Wrap-up and useful links

Serverless Definition

- Application logic (functions) is run in stateless compute containers
- These environments are event-triggered (async),
- ephemeral (may only last for one invocation),

Serverless Definition

So… There are actually Servers! YES! But...

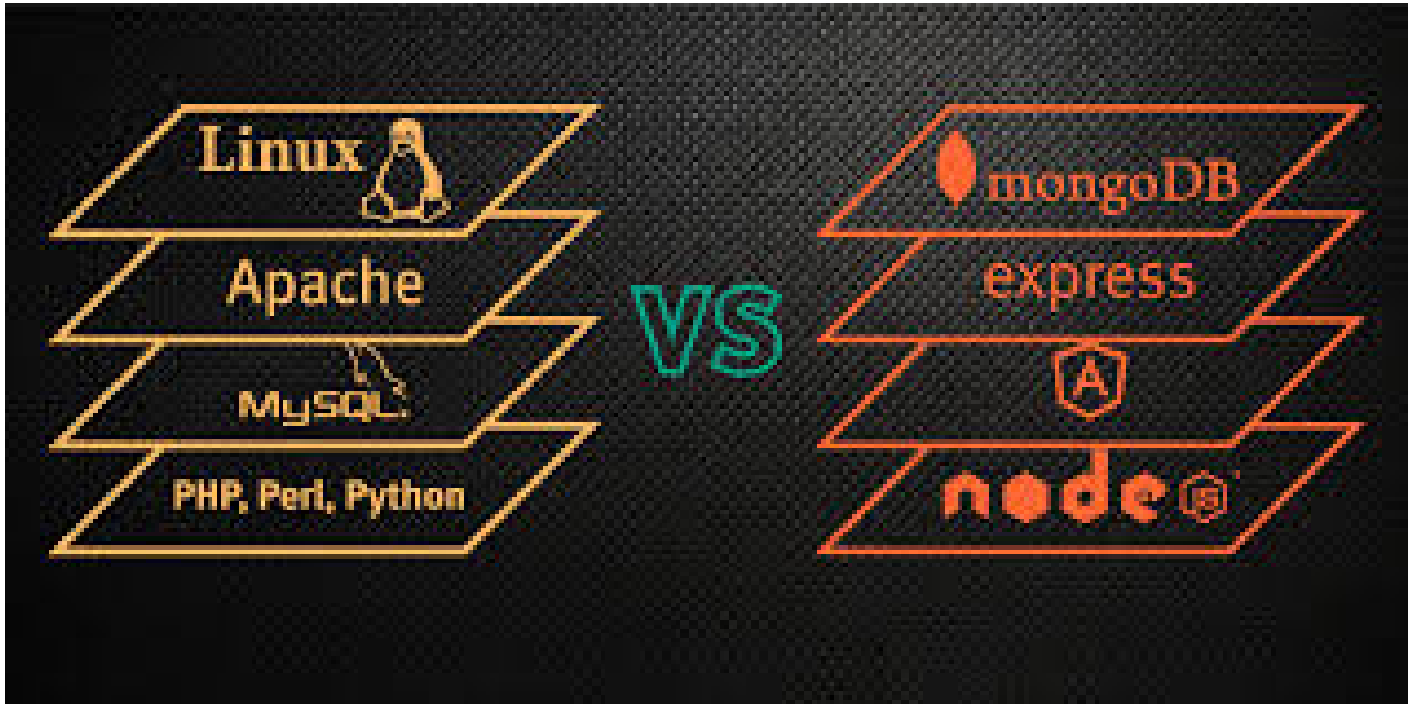
Fully **managed** by a cloud provider

Forget about scaling, pre-allocating, load-balancing…

Serverless and FaaS will be used as synonyms
(and sometimes Lambda functions as well)

Evolution of Web Stacks

- LAMPP/XAMPP vs MEAN… vs Serverless!!



https://www.infoworld.com/article/2937159/javascript/mean-vs-lamp-for-your-next-programming-project.html

Evolution of Cloud Platforms

Deploying an application to the Cloud involves a lot of decisions…

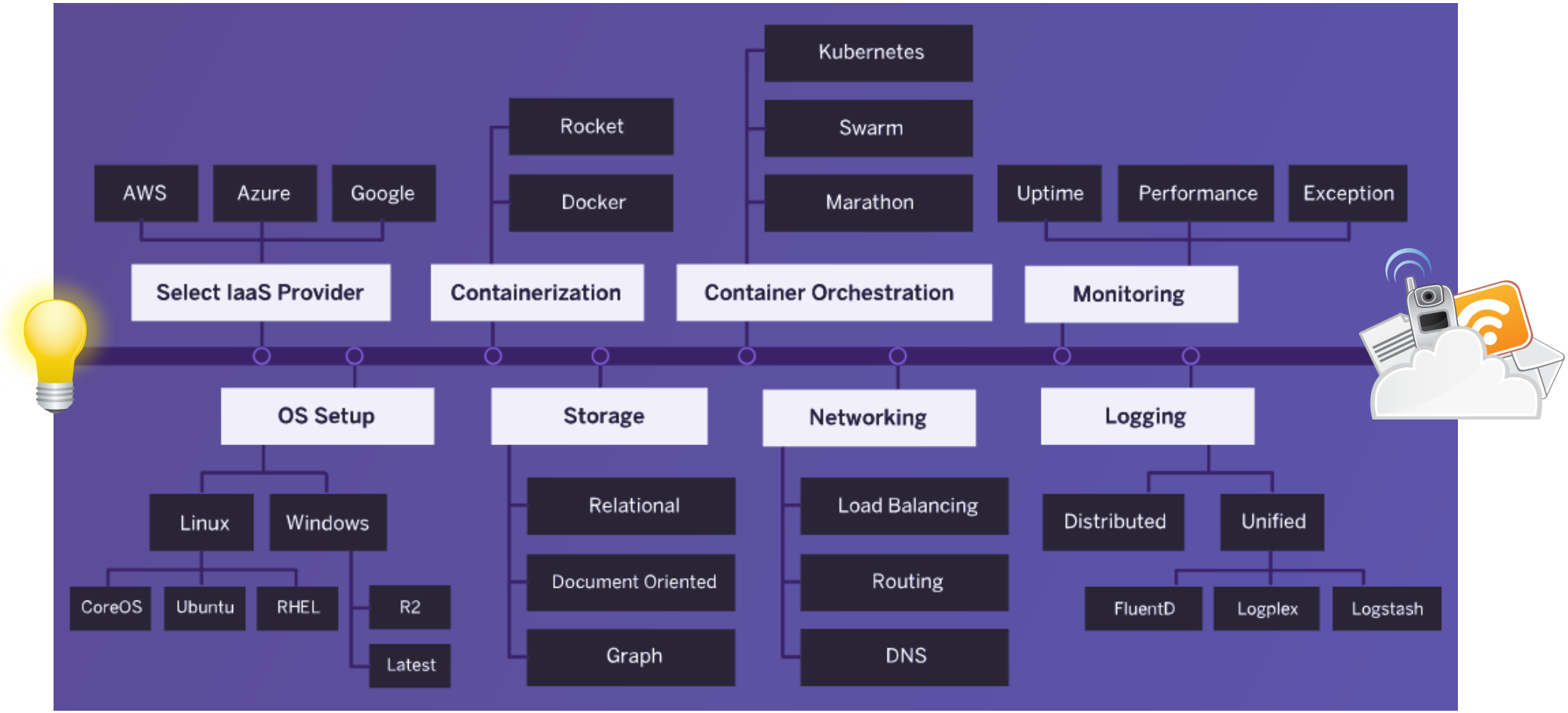
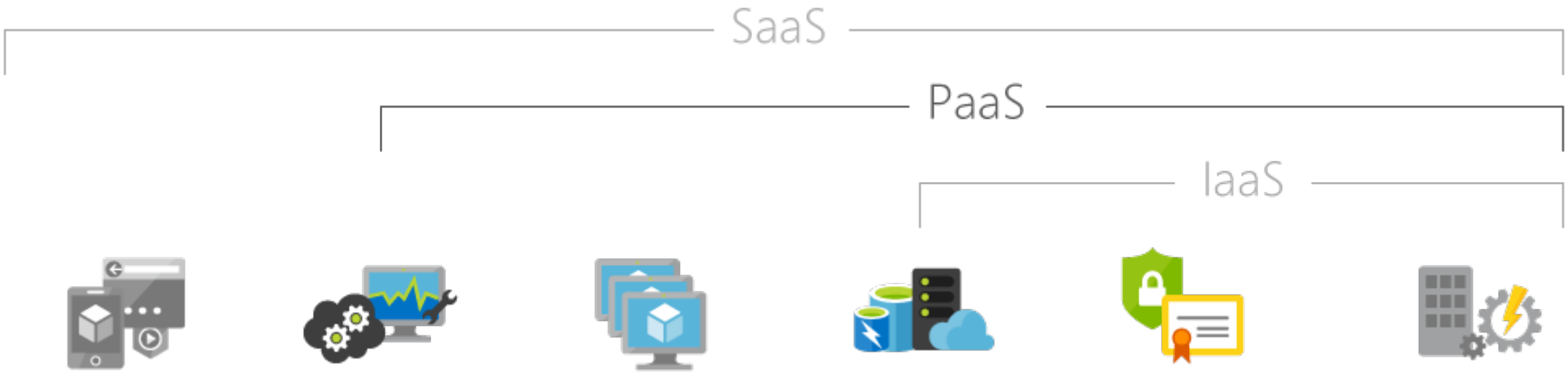


Image: Heroku – https://www.heroku.com/

Evolution of Cloud Platforms



Hosted applications

Development tools, database management, business analytics

Operating systems

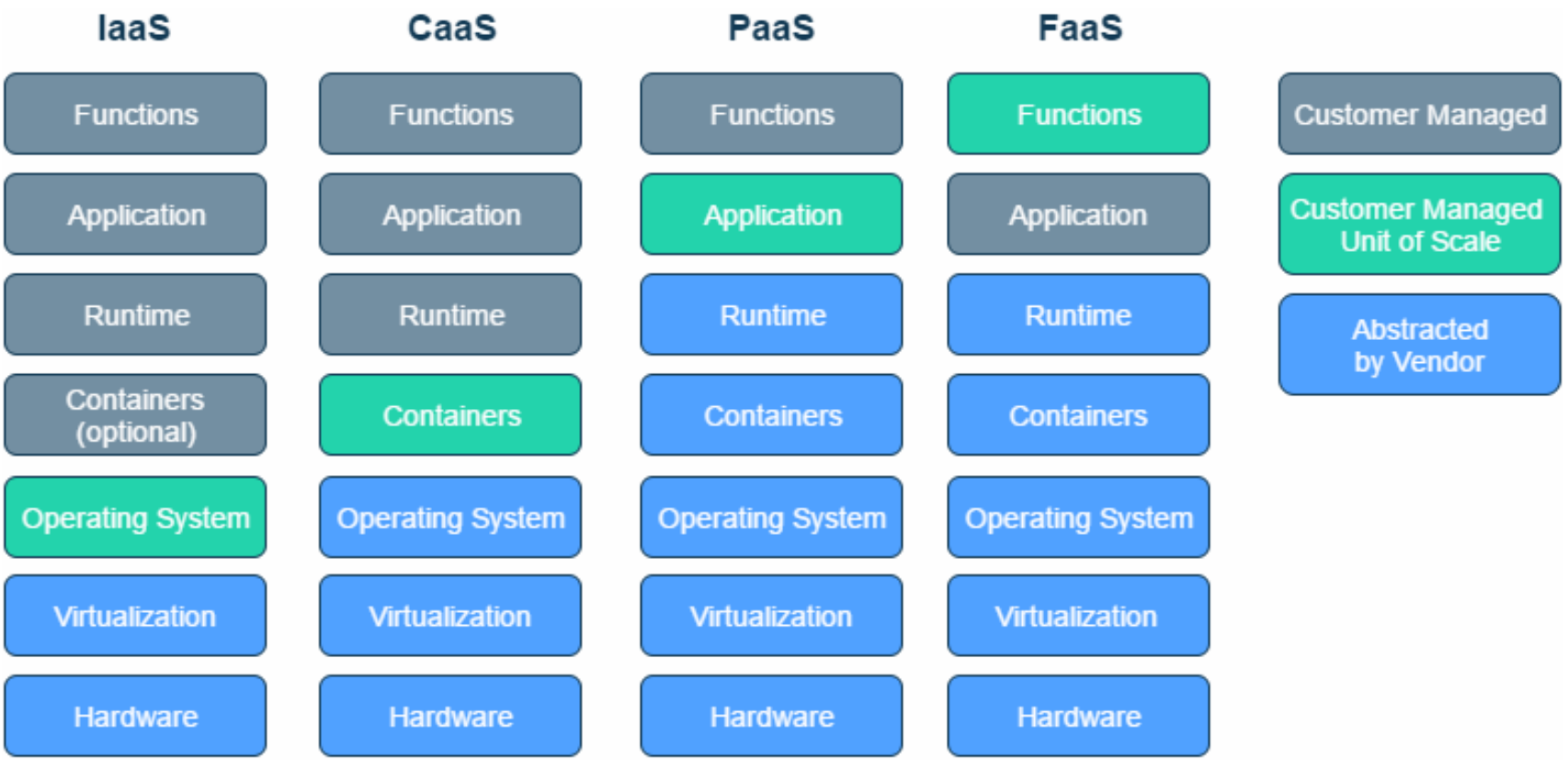
Servers and storage

Networking firewalls/ security

Data center physical plant/ building

Image: Azure – https://azure.microsoft.com/en-us/overview/what-is-paas/

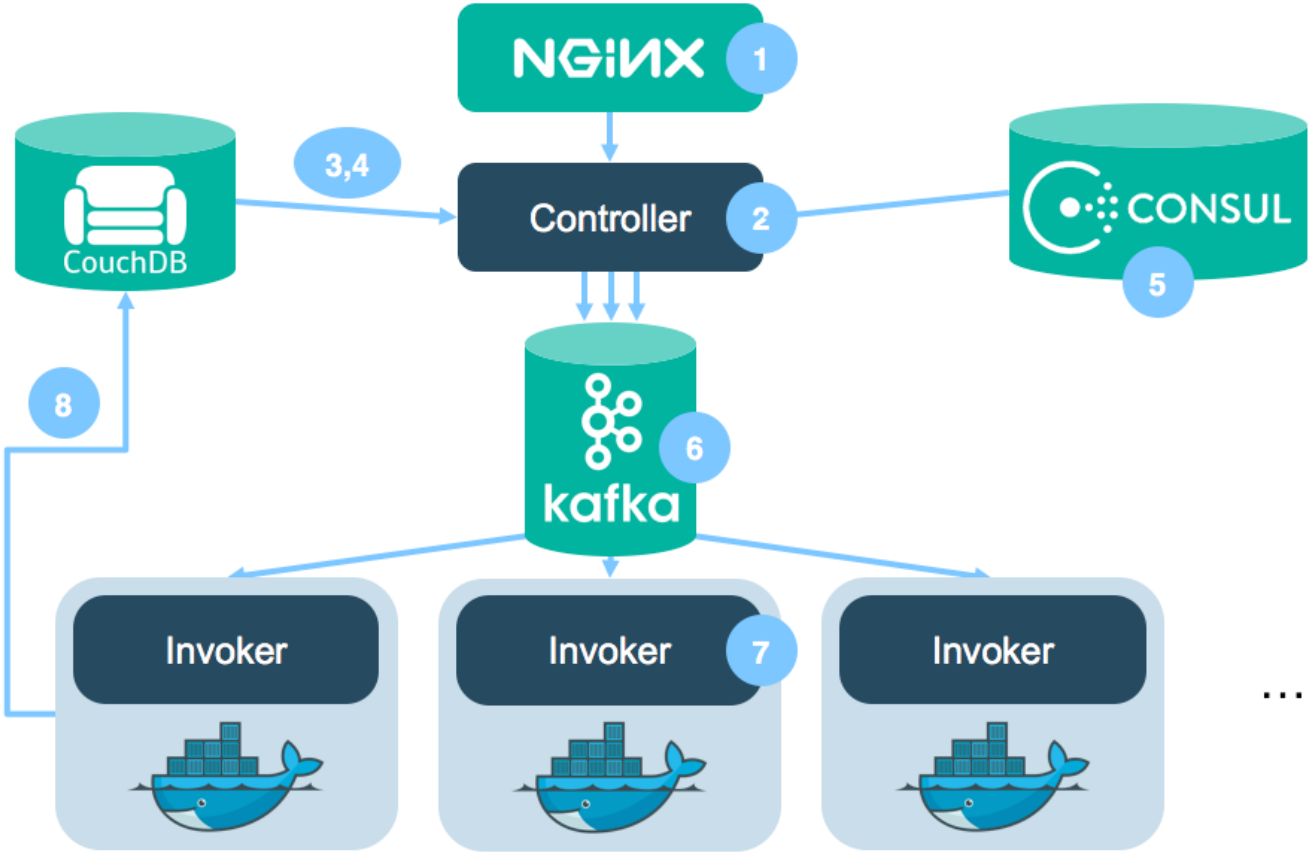
Evolution of Cloud Platforms



FaaS moves the unit of scale to **functions**:
Small pieces of application logic that run into a 3rd party managed platform (cloud)

Background

APACHE
OpenWhisk™



Evolution of Cloud Platforms

Provider	Languages
AWS Lambda	Node.js, Java, Python
Google Functions (Firebase)	Node.js
Azure Functions	Node.js, C#
IBM OpenWhisk (now Apache)	Node.js, Swift, Binary (docker)
Webtask.io	Node.js
Hook.io	Node.js, ECMAScript, CoffeeScript

Serverless Features

1) **Zero-management** of Server hosts or Server processes

2) Self **auto-scale** and **auto-provision**, based on load

3) **Costs** based on precise usage

4) **Performance** capabilities defined differently

5) Implicit **High availability**

Serverless Features

1) **Zero-management** of Server hosts or Server processes

Forget about:

autoscaling groups,

security groups,

load balancers…

There are servers... but we don’t

need to think about them anymore!

Serverless Features

2) Self **auto-scale** and **auto-provision**, based on load

Scale-up to serve any workload…

Scale-down to **zero** when the functions are not used

Concurrent execution limit to avoid DDoS attacks

Serverless Features

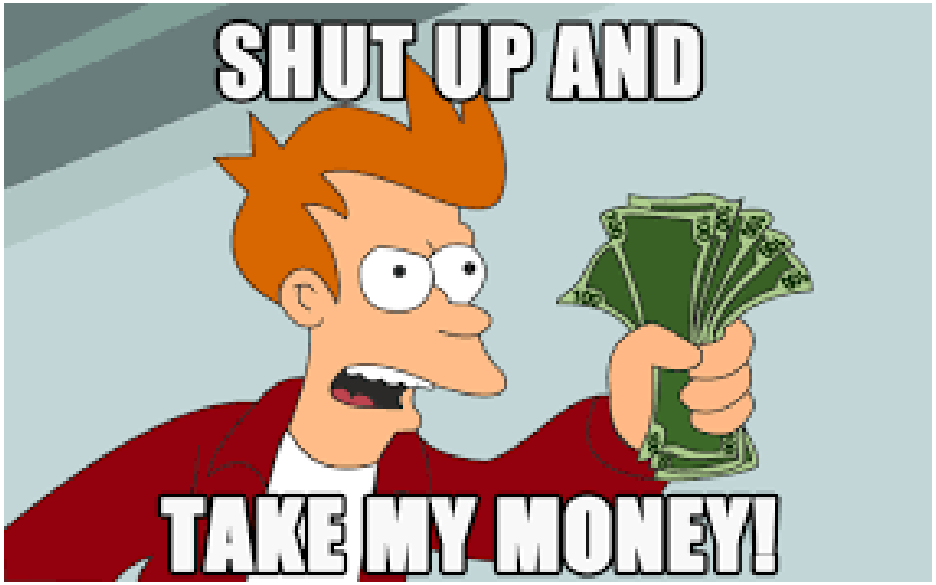
3) Costs based on precise usage

Pay-per-use (billed every 100ms)

Cost = Memory * Exec_time (ms)

More precise than hour/minute billing of VMS

No pre-allocation
of resources



Serverless Features

4) **Performance** capabilities defined differently

Forget about number / size of hosts

Memory is the only “knob” that can be tunned

vCPUs, networking and other resources are provisioned based on memory

Implicit **loss of control**

Serverless Features

5) Implicit **High availability**

Transparent, brought by cloud provider

No disaster recovery required

Less fault tolerance mechanisms

Wrap-up

Serverless is not a **free lunch**!

- Functions are **stateless**
- State should be managed somewhere else
- Vendor lock-in
- Testing Complexity

Wrap-up

But they represent the (r)evolution of **cloud** architectures!

- Built-in **Scalability**
- Fine-grained **Cost** model
- Reduced **time to market** and prototyping
- **Integration** with cloud PaaS

Useful Resources

martinfowler.com/articles/serverless.html

- Foundational article by Mike Roberts

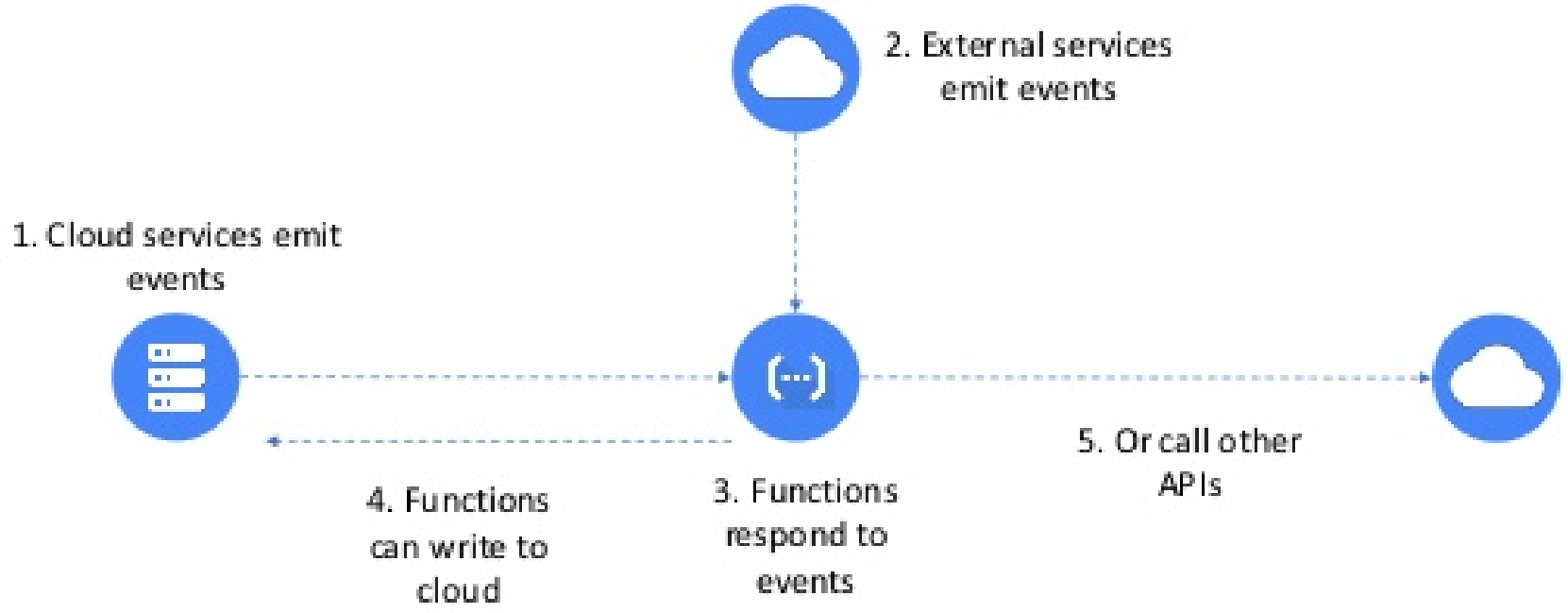
https://medium.com/@adrianco

- Adrian Cockcroft, the mastermind behind Netflix migration to microservices, now in AWS Lambda

github.com/JustServerless/awesome-serverless

- Curated list of resources related to serverless architectures and the Serverless Framework

Hands On!


2. External services emit events
1. Cloud services emit events
5. Or call other APIs
4. Functions can write to cloud
3. Functions respond to events

Hands On!

https://github.com/mgarriga/devfest-simple-functions

(Funciones helloworld y de prueba)

https://github.com/firebase/functions-samples

Generate-thumbnail function

Hands On!

```
npm install -g firebase-tools
firebase login
mkdir firebase-test
cd firebase-test/
firebase init functions
firebase use --add
firebase deploy --only functions
```

Crear un proyecto en blanco en firebase:
https://console.firebase.google.com/u/0/

Instalar npm y node.js 6 u 8
https://nodejs.org/en/download/package-manager/

Hands On!

File Edit View History Bookmarks Tools Help

working draft IoT - m | Firebase console | New Tab

https://console.firebase.google.com/u/0/

Firebase

Ir a la documentación

Te damos la bienvenida a Firebase

Herramientas de Google para programar grandes apps, interactuar con tus usuarios y ganar más a través de los anuncios móviles.

Más información | Documentación | Asistencia

Proyectos recientes

Agregar proyecto

Explorar un proyecto de demostración

cloud-functions-test
cloud-functions-test-7de74

Condiciones · Política de privacidad

Agregar un proyecto

Nombre del proyecto

cloud-functions-test2

Sugerencia: Los proyectos llevan las apps a distintas plataformas

ID del proyecto

cloud-functions-test2-178be

Ubicaciones

Estados Unidos (Analytics)
us-central (Cloud Firestore)

Utilizar la configuración predeterminada para el uso compartido de datos de Google Analytics for Firebase

- Comparte los datos de Analytics con Google para mejorar nuestros productos y servicios.
- Comparte los datos de Analytics con Google para habilitar la asistencia técnica.
- Comparte los datos de Analytics con Google para habilitar las comparativas.
- Comparte los datos de Analytics con los especialistas en cuentas de Google.

Acepto los Términos relativos a los controladores. Esto es obligatorio cuando se comparten datos de Analytics para mejorar los Productos y Servicios de Google. Más información

Cancelar | Crear proyecto

Hands On!

File Edit View History Bookmarks Tools Help 18:53

Recibidos - mgarriga cloud-functions-test

https://console.firebase.google.com/u/0/project/cloud-functions-test2-178be/functions/list 120% Search

Firebase

Project Overview

Desarrolla

- Authentication
- Database
- Storage
- Hosting
- Functions
- ML Kit

cloud-functions-test2 Ir a la documentación

Functions

Panel de control Estado Registros Uso

Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
helloWorld	HTTP Solicitud https://us-central1-cloud-functions-test2-178be.cloudfunctions.net/helloWorld	us-central1	Node.js 6	256 MB	60s

Panel de control Estado Registros Uso

Registros de bús... Todas las funciones Todos los niveles de registros

Hora	Nivel	Función	Mensaje del evento
14 sep. 2018			
6:50:47.513 p.m.		helloWorld	Function execution took 179 ms, finished with status code: 304
6:50:47.335 p.m.		helloWorld	► Billing account not configured. External network is not accessible and quotas are severely limited. Configure billing account …
6:50:47.335 p.m.		helloWorld	Function execution started

Thanks!!!

Serverless Architectures: The (r)evolution of Cloud Resource Management

Martin Garriga – martin.garriga@fi.uncoma.edu.ar