



# Beyond Containers & VMs

An Introduction to Serverless Application Architecture

Ali Ghorbani Moghadam  
Technical Leader, Cisco

*Why Serverless?*

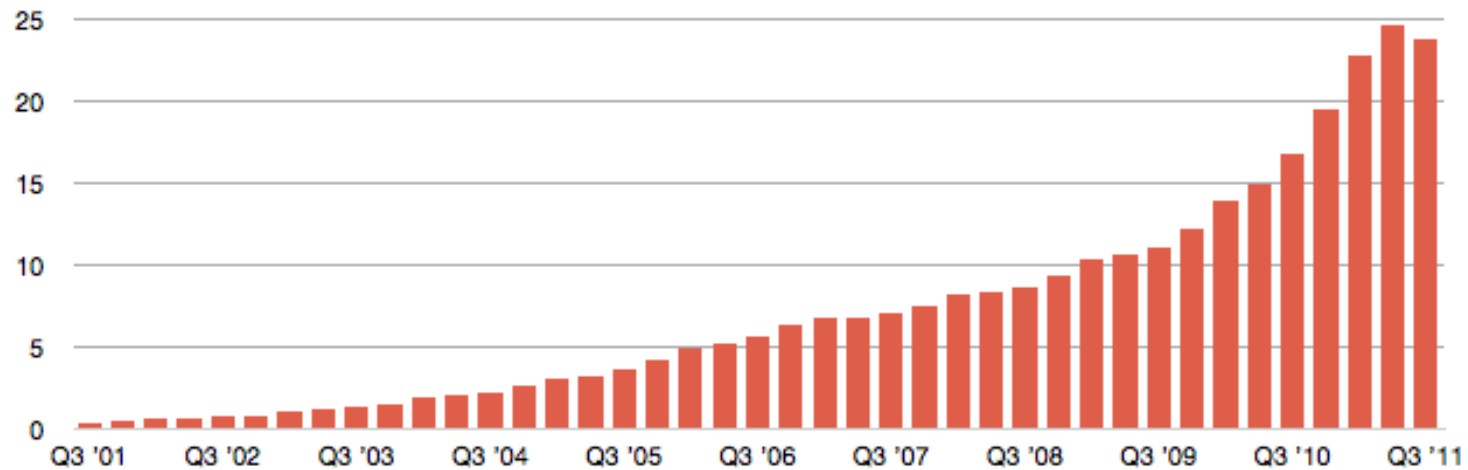
# KitchenAid®



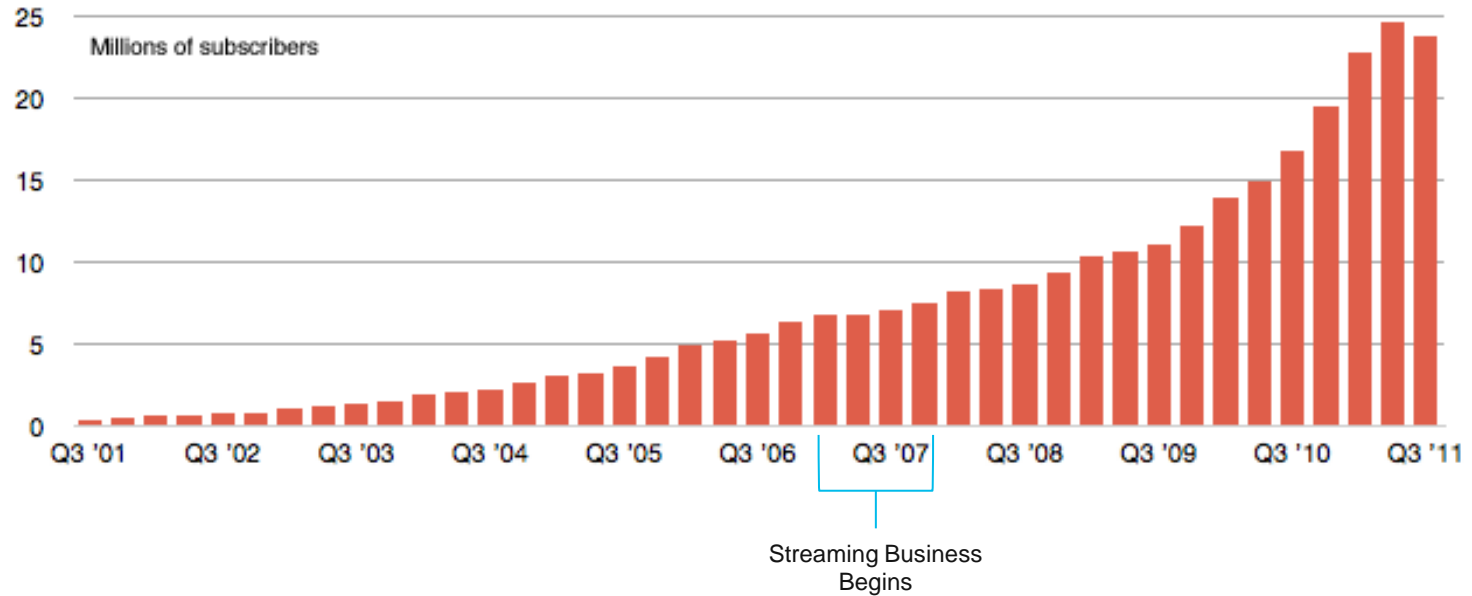
<https://www.kitchenaid.co.uk/Support/Where-to-buy>

Source: <https://twitter.com/gabriperego/status/1004077611335782400>

# What's This?



# First 10 Years Subscriber Growth



# Who Led This? Adrian Cockcroft



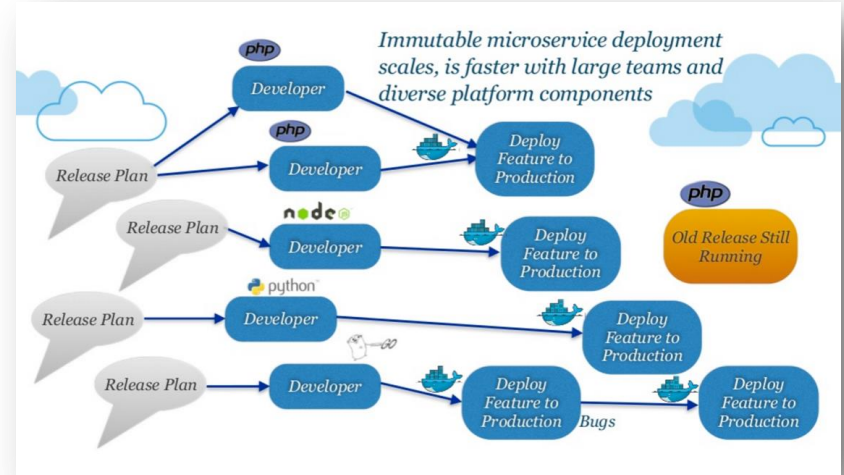
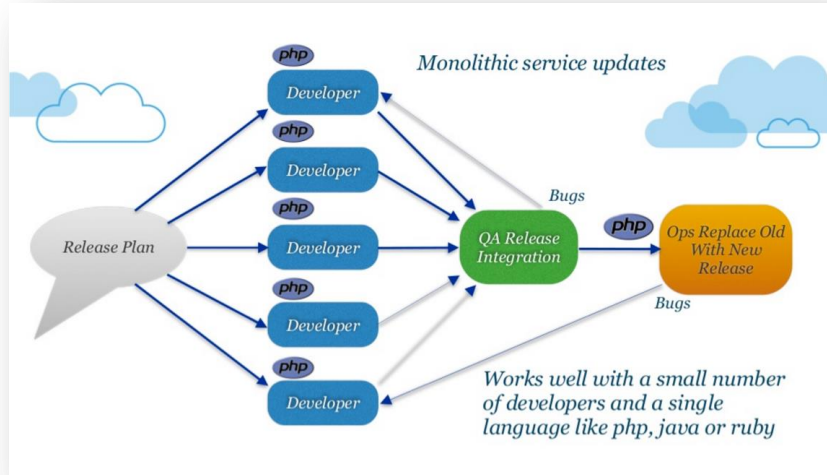
- Netflix Director of Web Engineering
  - 2007-2010
- Netflix Cloud Architect
  - 2010-2014
- Battery Ventures Technology Fellow
  - 2014-2016
- AWS VP Cloud Architecture Strategy
  - 2016- Present

# Adrian's Battery Venture pitch began with . . .

## What I learned from my time at Netflix

- *Speed wins in the marketplace*
- *Remove friction from product development*
- *High trust, low process, no hand-offs between teams*
- *Freedom and responsibility culture*
- *Don't do your own undifferentiated heavy lifting*
- *Use simple patterns automated by tooling*
- *Self service cloud makes impossible things instant*

# Adrian: Monolithic vs Microservices



All In The Name of More Iterations, More Innovation



# Why Serverless?

**More Iterations,  
More Innovation**

# Application Architecture Approaches

- Given time to create a new unit of compute



Physical Servers  
(Months)



Virtual Machines  
(Minutes)



Containers  
(Seconds)



Function-as-a-Service  
(Milliseconds)



**Pets/Mode 1/Monoliths**  
Go to great lengths to  
keep compute alive



**Cattle/Mode 2/Microservices**  
Create and destroy compute  
frequently



**Serverless**  
Smaller and less coupled

# *How Serverless Works*

# Some Terminology & Technology Maturity

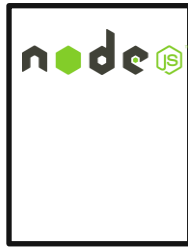
**Serverless** = The application architecture approach

**FaaS** = The underpinnings that make it possible

Serverless is to FaaS as Microservices are to Containers

Serverless 2019  $\approx$  Cloud 2011

# How FaaS Runtimes Work

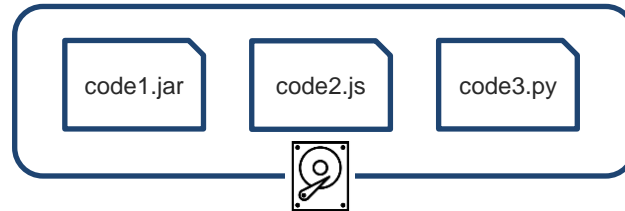


Standby containers  
w/ language runtimes  
but no app code



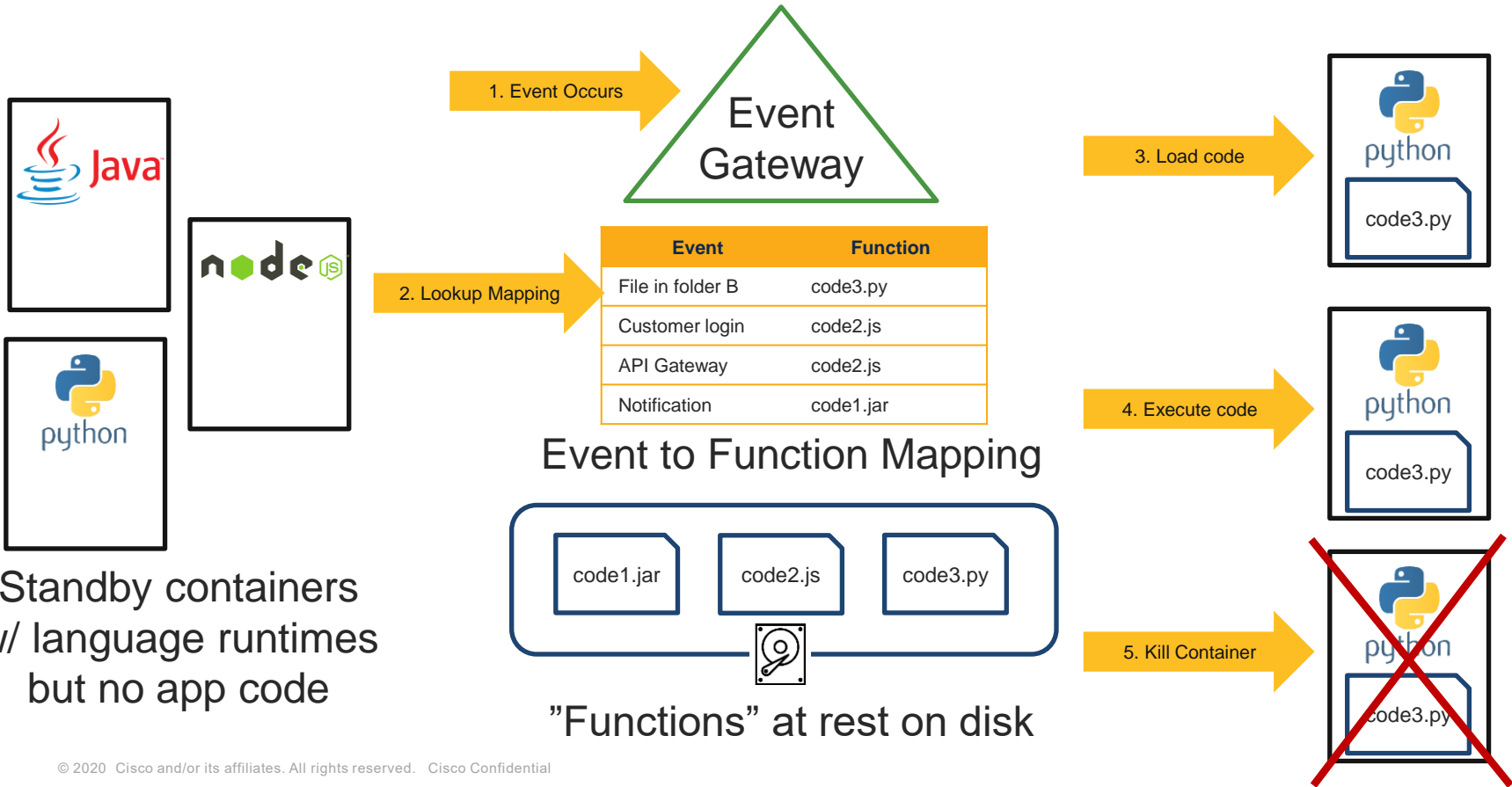
Event	Function
File in folder B	code3.py
Customer login	code2.js
API Gateway	code2.js
Notification	code1.jar

Event to Function Mapping



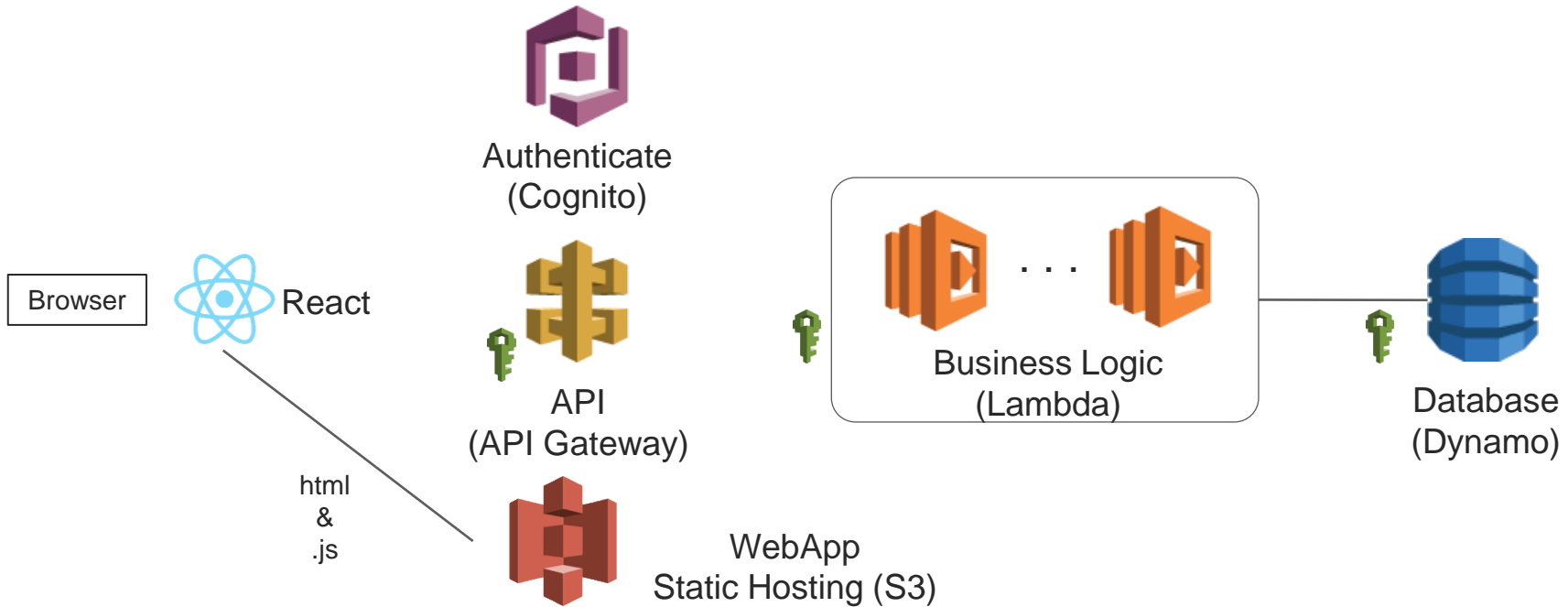
"Functions" at rest on disk

# How FaaS Runtimes Work



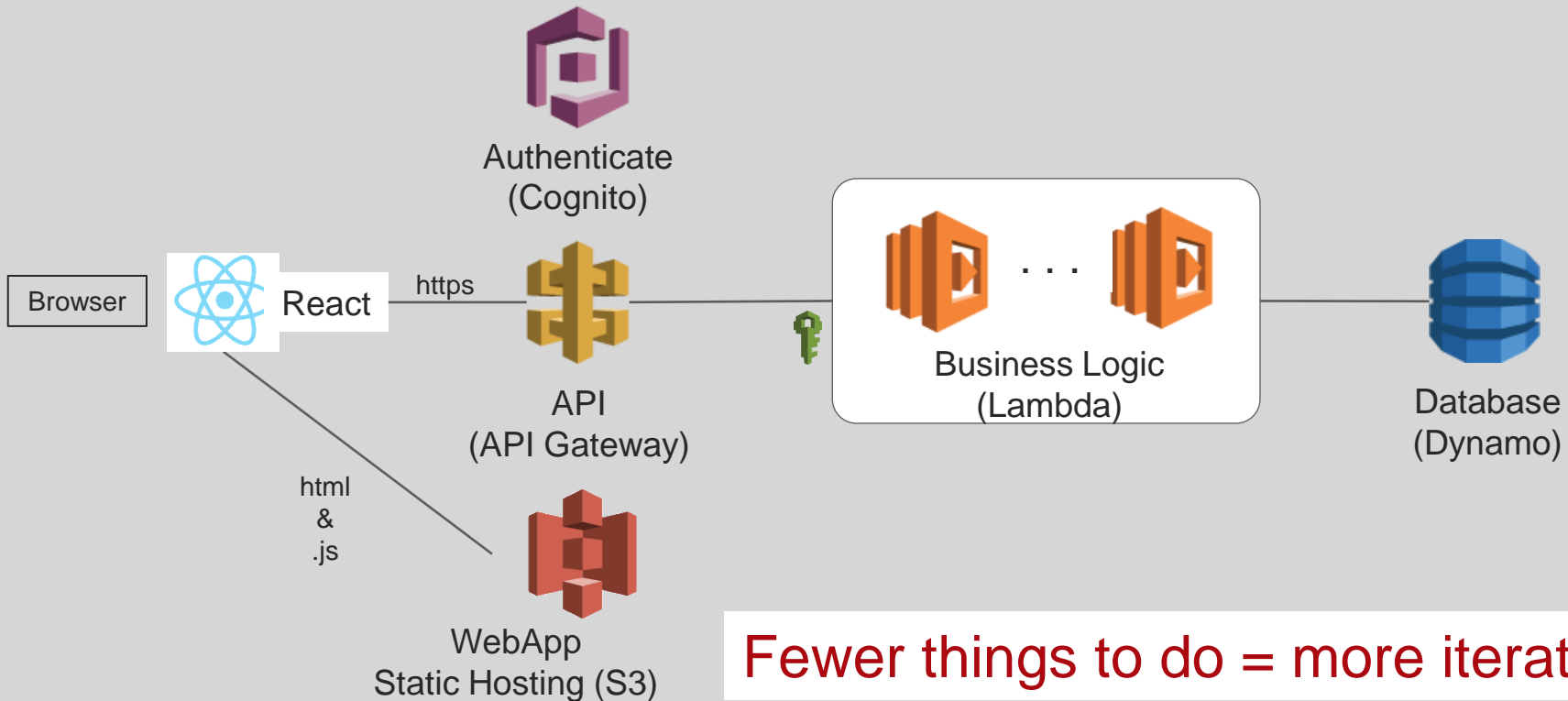
# *Serverless Services & Service Ecosystem*

# Sample Serverless Application Architecture





# Sample Serverless Application Architecture



**Fewer things to do = more iterations!**

# *Considerations of a Serverless Approach*

# Pros & Cons

- Pros

- Fewer pieces of infrastructure to procure
- Quicker and more iterations leads to more innovation and faster time to market
- Easy to get started (automate devops tasks)

- Cons

- More difficult to debug
- Ops doesn't go away, even if you think it does
- Easier to get locked into a particular cloud
- Young ecosystem

