



What's All the Fuss About Serverless?

May 13th, 2019



Hello!
I'm Taylor Krusen.
I work for Dropbox.

Let's talk about
Serverless

Twitter: @TaylorKrusen

Overview

What I'll cover

- Serverless as a concept
- Reasons for popularity
- Pragmatic usage

Who is this talk for?

Developers with...

- Curiosity / interest in serverless
- Limited or no exposure to serverless

Goals

- Understand serverless and the situations where it will benefit you
- Look past marketing jargon
- Navigate the ecosystem of tools
- Get excited about serverless



Tweet to Sheet



A screenshot of a Google Sheet with a dark blue header and a light grey toolbar. The sheet contains a single row of data representing the tweet. The columns are labeled "Tweet" and "Name". The "Tweet" column contains the text "Hello #iJS19! Want to learn about Serverless? Join me today at 17:05 in Albert 1/2! <https://t.co/EiqWc878JP>". The "Name" column contains the text "Taylor Krusen".

	Tweet	Name
1	Hello #iJS19! Want to learn about Serverless? Join me today at 17:05 in Albert 1/2! https://t.co/EiqWc878JP	Taylor Krusen

Twitter: @TaylorKrusen

Tweet to Sheet Master Plan



- Webhook fires on account activity



- Code runs on AWS Lambda



- Specific tweets written to sheet

Participate?!

- Tweet @TaylorKrusen and include the hashtag #Jokes
- Your tweet is added to sheet. Review list at end.



Serverless

Popularity

● serverless
Search term

+ Compare

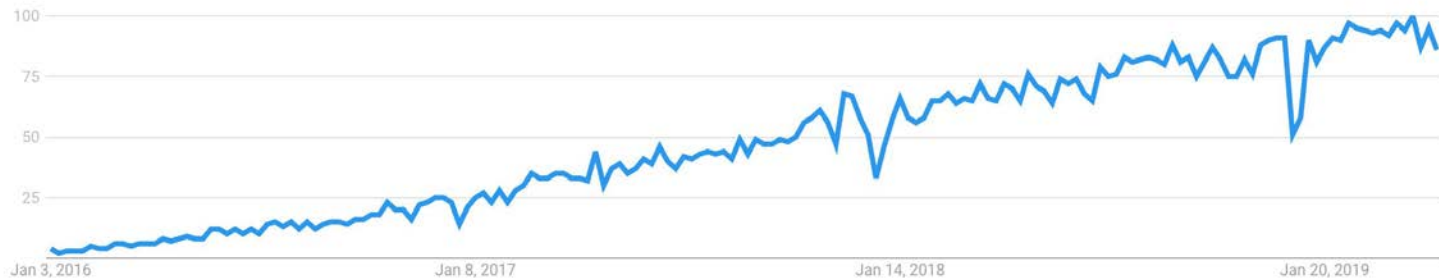
Worldwide ▼

1/1/16 - 5/1/19 ▼

All categories ▼

Web Search ▼

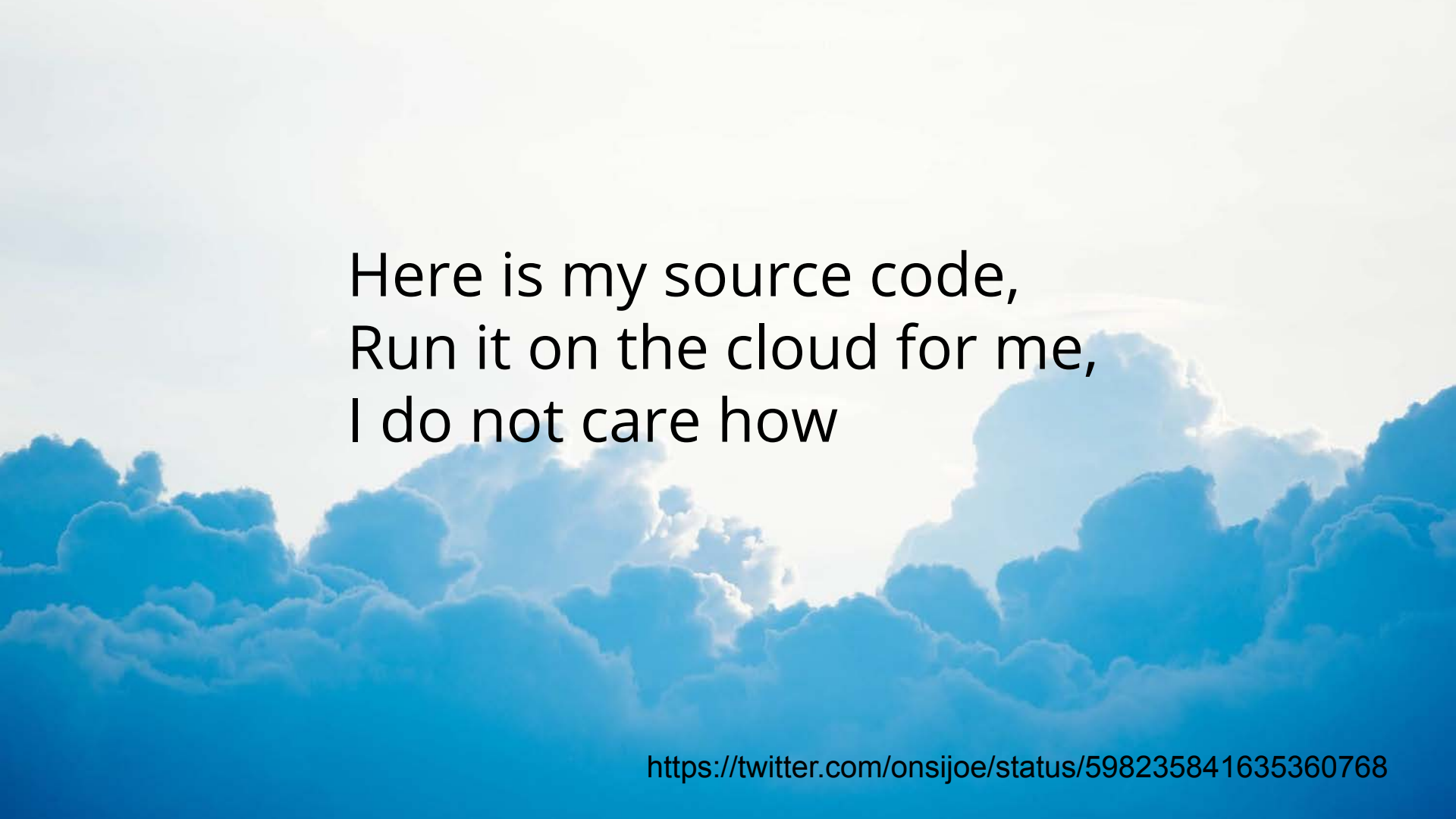
Interest over time ?





DUDE

WHERE'S MY SERVER



Here is my source code,
Run it on the cloud for me,
I do not care how

<https://twitter.com/onsijoe/status/598235841635360768>

Serverless



Abstraction

- ✓ Don't need to own or provision a server.

Event-driven

- ✓ Managed FaaS in the cloud.

Pay-per-use

- ✓ Only charged for code that runs

Serverless Spectrum

Less
Serverless



More
Serverless

Degree of *serverlessness*

- Reliance on BaaS (third-party services)
- Ephemeral computing
- Degree of 'control' over server
- Coupling of resources used and resources billed

'The Serverless Spectrum' by Ben Kehoe

“

“Abstraction is selective ignorance”

- Andrew Koenig



Alphabet Soup

- **FaaS = Function as a Service**

- Allows users to develop, run and manage app functionalities without building or maintaining the related infrastructure.

- **BaaS = Backend as a Service**

- Middleware that allows developers to connect their app to cloud services.

- **PaaS = Platform as a Service**

- Similar to FaaS, but different architecture and scaling.
- Long running application thread.
- Bill per time running rather than by execution.

- **IaaS = Infrastructure as a Service**

- Hardware is provided and managed by an external vendor.

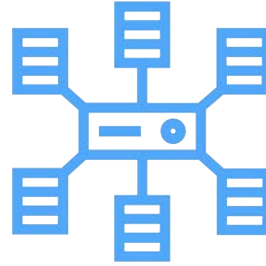
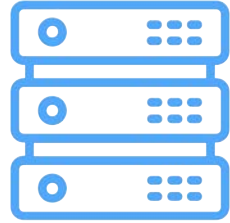
- **Ephemeral**

- Something short-lived or temporary.

- **Server**

- A computer device or program that provides functionality for other programs / devices.





On-Prem

IaaS

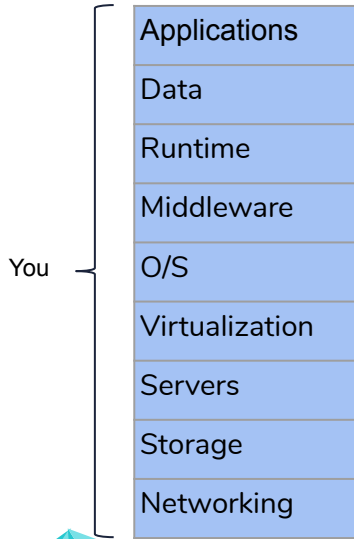
PaaS

Serverless

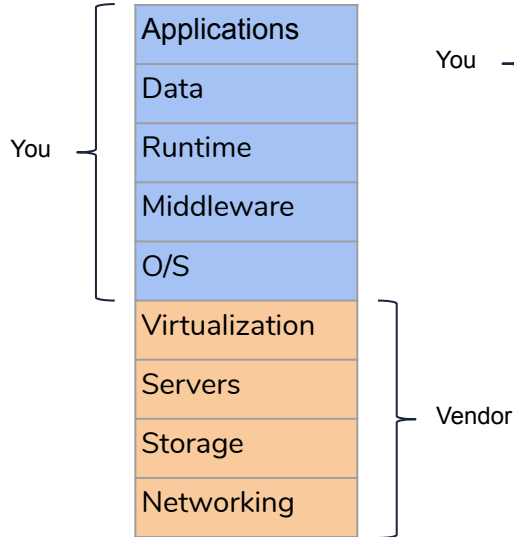


Evolution of Cloud Offerings

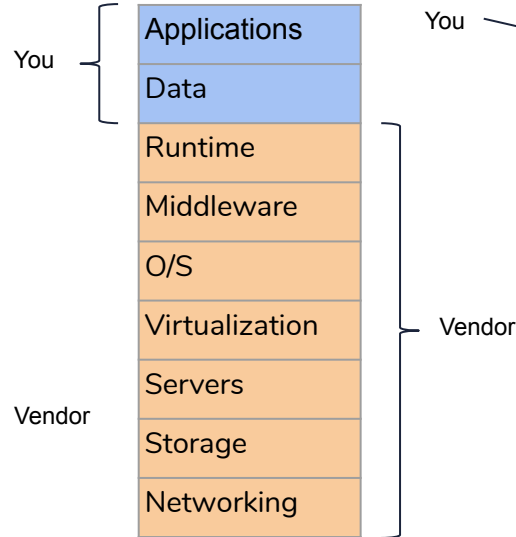
On Premise



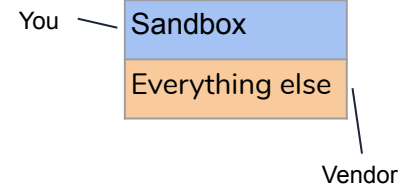
IaaS



PaaS



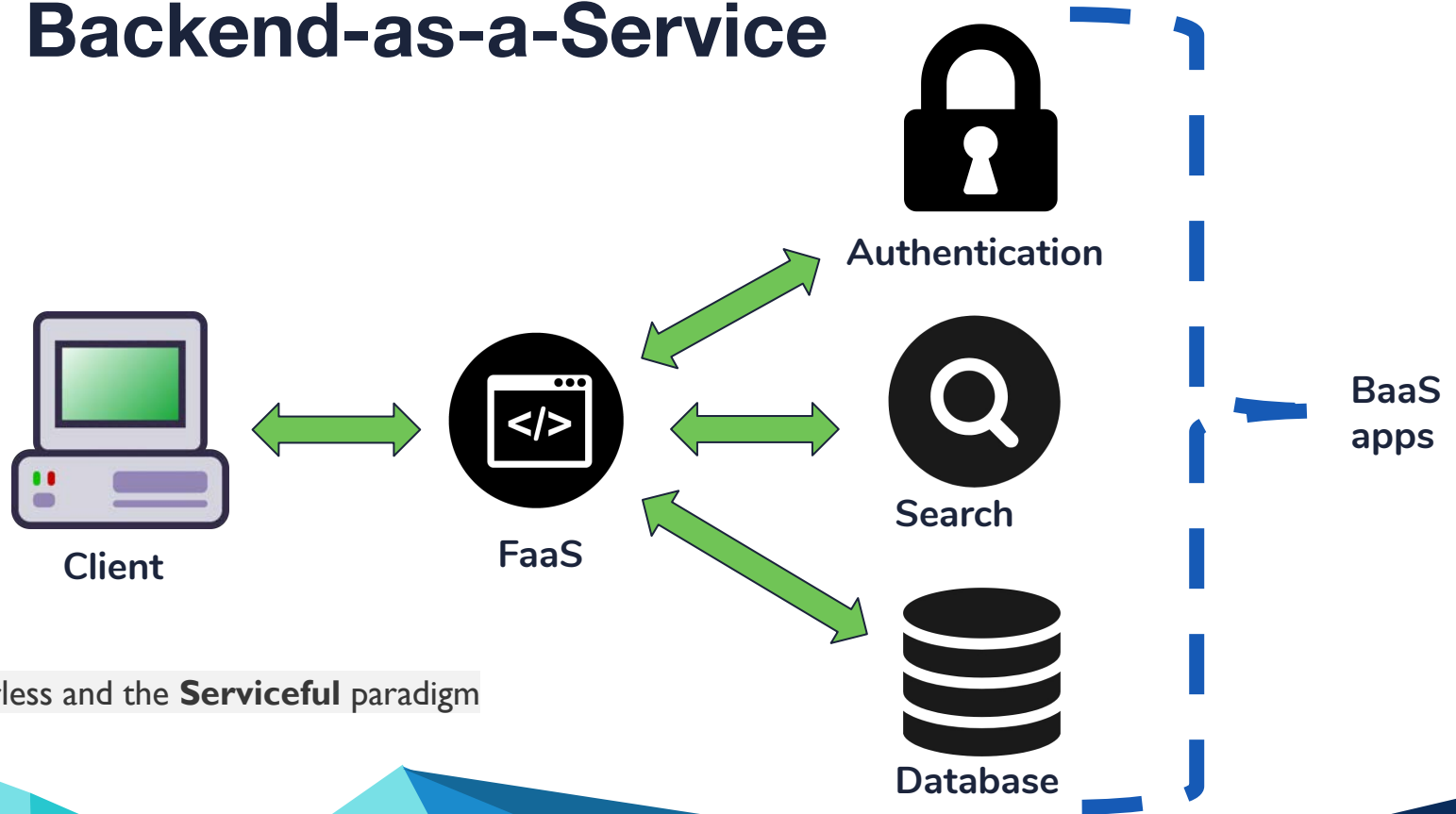
Serverless



BaaS

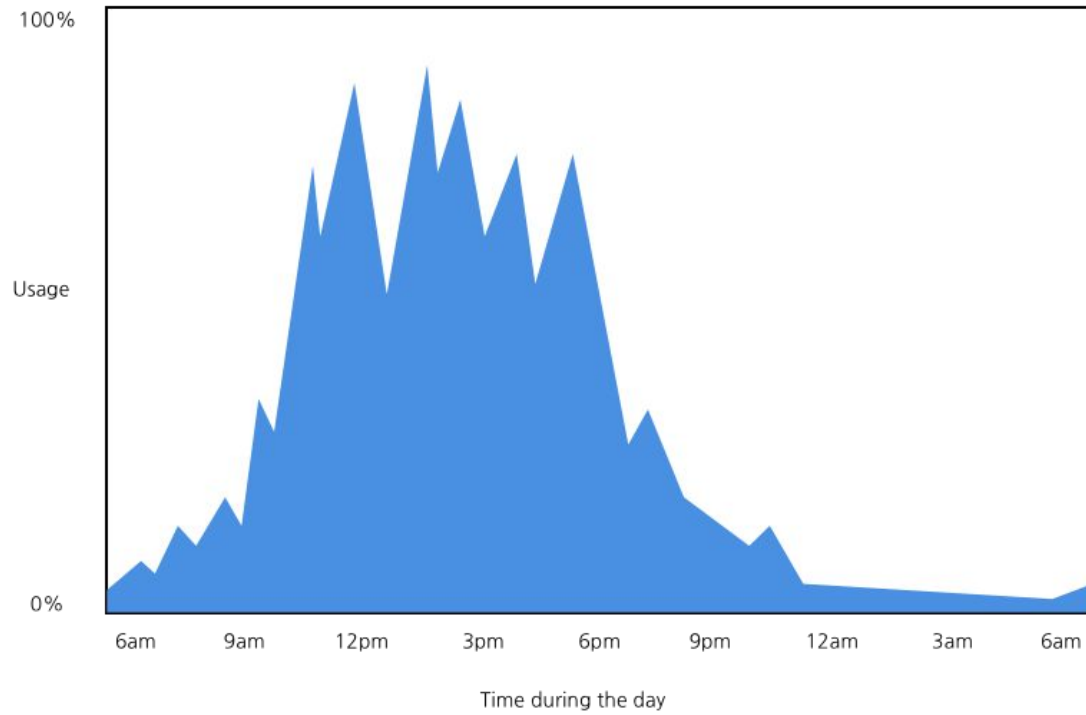
Buy instead
of Build

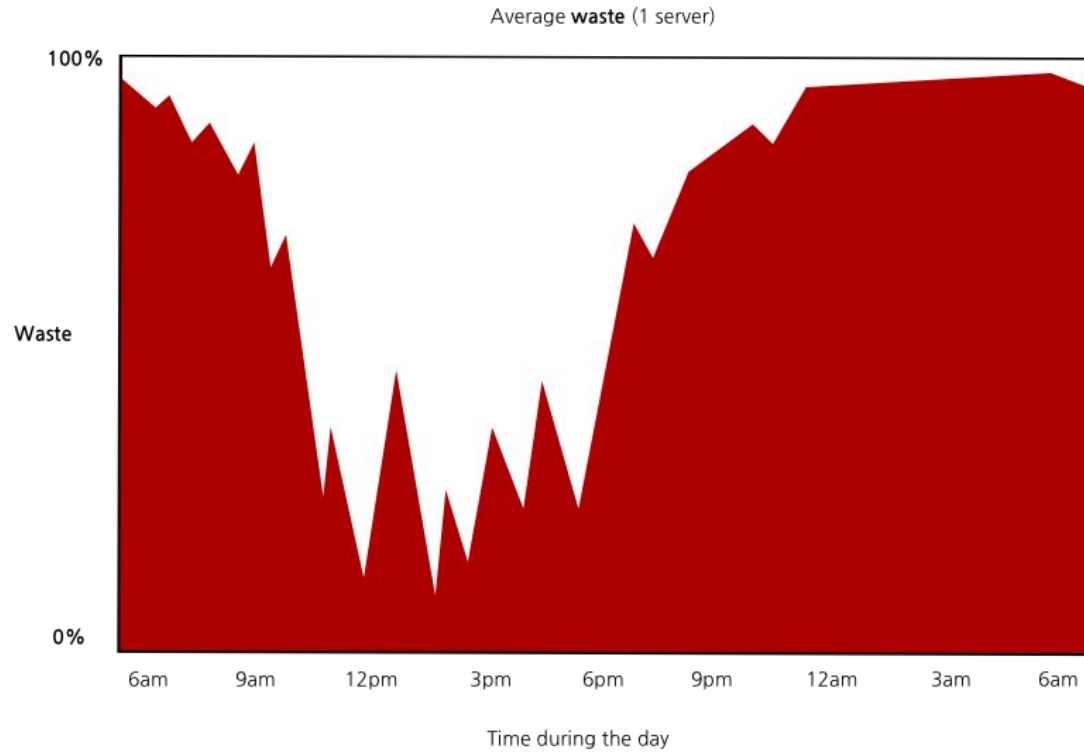
Backend-as-a-Service



Serverless and the **Serviceful** paradigm

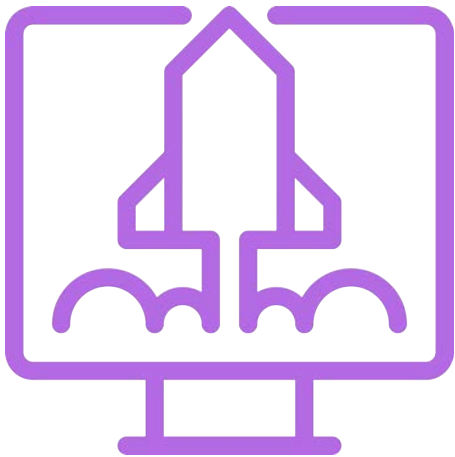
Average usage (1 server)



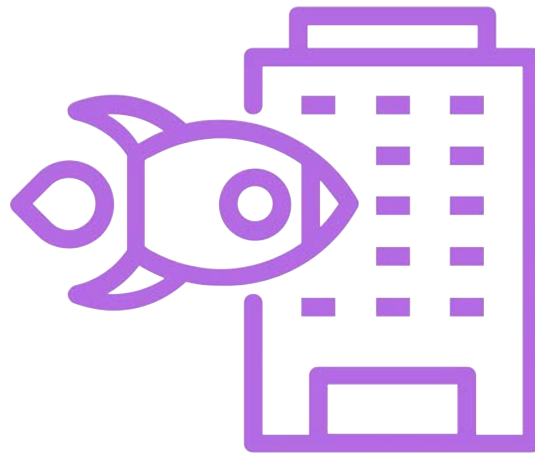




Winners

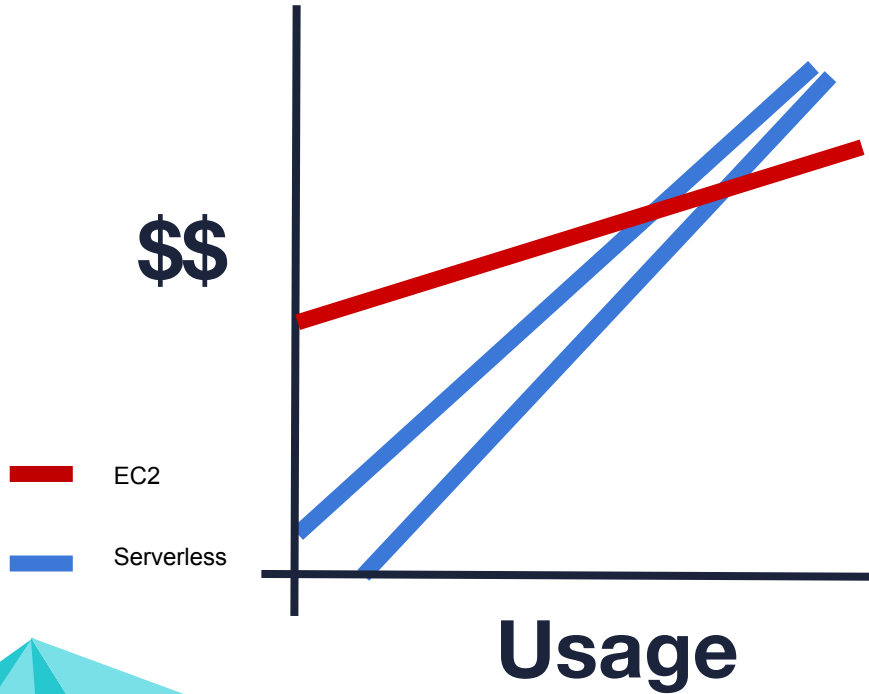


Startup and
small business



Enterprise

Why



- Developers can focus on business value
- Auto-scaling web apps and APIs
- Disruptive pricing model

<https://www.bbva.com/en/economics-of-serverless/>

Benefits and Compromises

Speed / Velocity / Agility

- Faster time to market
- Less to build

Simplicity

- Very easy for users of the FaaS

Stateless

Lack of tooling

Less control

- No knobs to tweak

Architectural complexity

- 'mini monoliths'
- Someone needs to wrap their head around everything



Benefits and Compromises

Lower operational burden

- Outsourced infrastructure
- Fewer people
- 'Better' security and reliability

Implementation drawbacks

- Integration testing
- Versioning / packaging
- May need separate FaaS for everything

Reliance on 3rd party tools

- Effectiveness
- Reliability
- Vendor lock-in
- Risk

Benefits and Compromises

“

Serverless is a way to focus on business value.

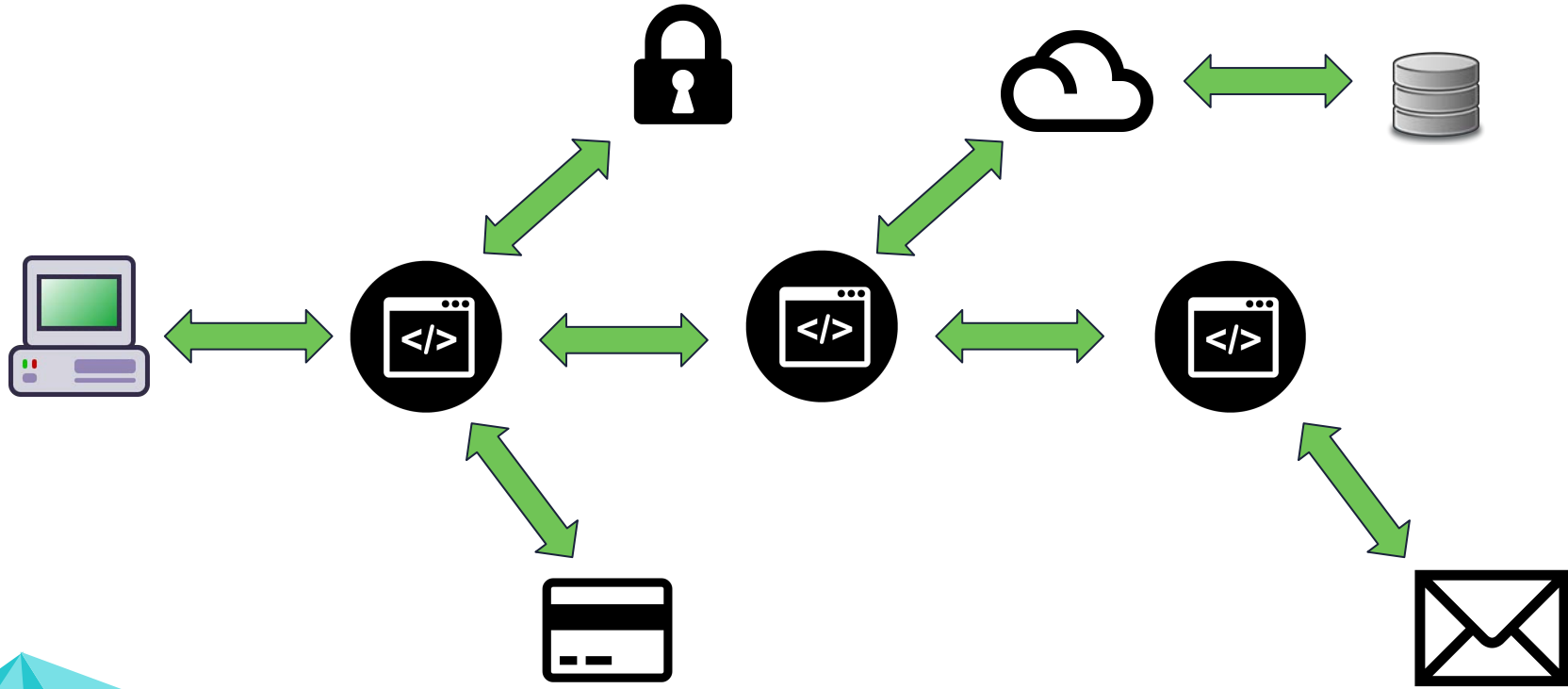
~ Ben Kehoe -- Serverless is a State of Mind





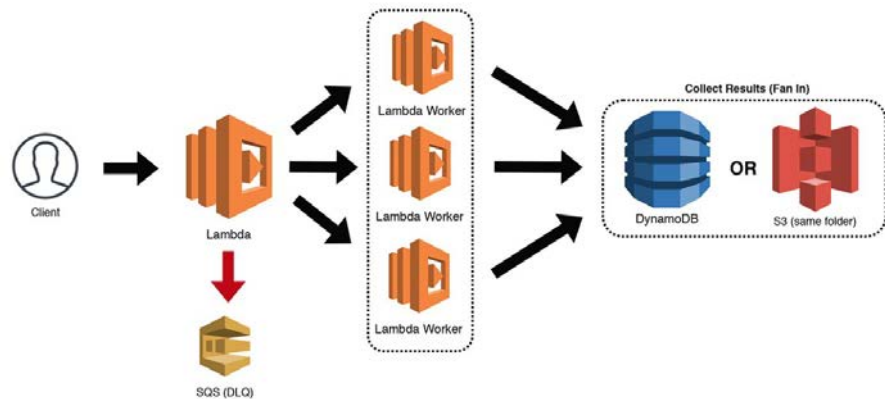
Shifting Paradigms

Serverless Architecture



Building Differently

'Serverless Microservice Patterns for AWS' by Jeremy Daly




Fan-out/Fan-in Pattern

<https://www.jeremydaly.com/serverless-microservice-patterns-for-aws/>

Use Cases

Scripts triggered by events

- Your custom code reacts to 'events'.
 - Cron job: trigger functions on a schedule.
- 
- External: web hook
 - Internal: closed ecosystem like Lambda and an s3 event

Web applications

- UI driven application calling HTTP endpoints that trigger your code

Schneider Electric





Serverless Economy

Serverless Web Apps

A web app is more than FaaS...



Typically consists of:

- Lambda
- API Gateway (HTTP endpoints)
- S3 to serve static content
- DynamoDB
- Many others...

Hidden Costs

TimerCheck.io

- Over 2m requests
- 300k+ seconds of compute

Details	Total
AWS Service Charges	\$11.12
▶ API Gateway	\$7.47
▶ CloudTrail	\$0.00
▶ CloudWatch	\$1.51
▶ Data Transfer	\$0.04
▶ DynamoDB	\$0.00
▶ Elastic Compute Cloud	\$0.73
▶ Lambda	\$0.22
▶ Route 53	\$1.09
▶ Simple Notification Service	\$0.00
▶ Simple Storage Service	\$0.07

Major Players



- Serverless offering: **Azure Functions**
- Launched in March, 2016



- Serverless offering: **Cloud Functions**
- Launched in Dec, 2017
- Available as 'OpenWhisk' in Dec, 2016



- Serverless offering: **Cloud Functions**
- Launched in March, 2017
- General Availability on July 24, 2018



- Serverless offering: **Lambda**
- Launched in Nov, 2014
- Most mature ecosystem

Cloud Fight

The Future?

Lambda







- Runs on Linux environment
- Functions built as standalone elements
- Provisions memory *per function*
- Better scaling for HTTP endpoints

Azure Functions



- Runs on Windows environment
- Multiple functions grouped together as an application
- Provisions memory *per application*
- Platform is *very* user friendly
- Robust developer resources

Serverless Providers

	AWS (\$)	Microsoft (\$)	Google (\$)	IBM (\$)
				
L	0.00	0.00	0.00	0.00
M	18.55	4.40	9.76	3.83
H	799.76	603.40	709.95	630.70
!!	22,667.13	20,093.40	23,321.20	21,243.20

L = 1,000ms & 128mb &
1m executions





M = 1,000ms & 128mb &
5m executions

H = 3,000ms & 256mb &
50m executions

!! = 5,000ms & 512mb &
500m executions

Estimates via serverlesscalc.com from
@acloudguru

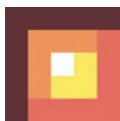
Supported Languages

Language	Amazon	Microsoft	Google	IBM
				
Node.js	Y	Y	Y	Y
Python	Y	Partial	Partial	Y
Java	Y	N	N	Y
C#	Y	Y	N	Y
Go	Y	N	N	Y
F#	N	Y	N	Y
Swift	N	N	N	Y
PHP	N	Partial	N	Y

Serverless Providers

Other points of consideration?

- Your specific needs
- Ecosystem
- Community



Auth0's Webtask



Oracle's Fn Project



Apache's OpenWhisk



APACHE
OpenWhisk™

- Any language!!
- Open source serverless platform
- Can run locally out of a container
- Choose your cloud (or host it yourself)
- Reusable and extensible
- Good introduction to distributed systems



Serverless Framework

Open-source CLI for building serverless architectures. At 22,000 stars on GitHub, the Serverless Framework started a movement.

 ★ 22,373  5,000  1,700  5.3M deploys

Deploy your serverless code to:

- AWS Lambda
- Azure Functions
- Google Cloud Functions
- IBM Cloud Functions
- Others...



State of Serverless

Steadily Moving Forward

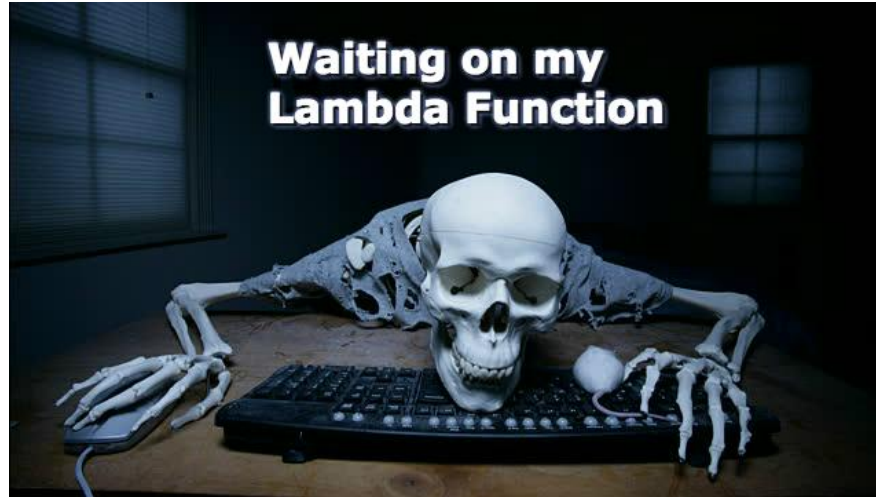
(Can we please stop talking about AWS now?)

Lambda



- Lambda Runtime API
- Lambda Layers
- Websocket support in API Gateway & Lambda
- AWS IDE integration
- AWS Firecracker goes open-source
- Aurora
- DynamoDB on demand
- Timestream timeseries database

Cold Starts



<https://mikhail.io/2018/08/serverless-cold-start-war/>



chrismunns

@chrismunns

Follow



OK [#serverless](#) [#awslambda](#) friends. Want to know how to *properly* do pre-warming of Lambda functions? [@jeremy_daly](#) has codified our best practices for it right here: [github.com/jeremydaly/lam....](https://github.com/jeremydaly/lam...) NOTE: you *may* not need this at all! don't prewarm just cause!



jeremydaly/lambda-warmer

A module to optimize AWS Lambda function cold starts -
jeremydaly/lambda-warmer

github.com

7:25 AM - 13 Jul 2018

43 Retweets 109 Likes



4



43



109



- Task: calculate all prime numbers less than 1,000,000.

Memory Allocation	Execution Time	Cost
128 MB	11.72296 sec	\$0.024628
256 MB	6.67894 sec	\$0.028035
512 MB	3.194954 sec	\$0.026830
1024 MB	1.46598 sec	\$0.024638

<https://www.slideshare.net/ChrisMunns/aws-startup-day-boston-2018-the-best-practices-and-hard-lessons-learned-of-serverless-applications>





What

- Open specification about event metadata

Who

- Support from IBM, Google, Red Hat, many more
- First class support from Microsoft Azure

Why

- Interoperable cloud architectures
- Distributed data across vendors and clouds

The Serverless and Event-Driven Future
<https://www.youtube.com/watch?v=TZPPjAv12KU>

Questions?

... and #Jokes Review!

Twitter: @TaylorKrusen

