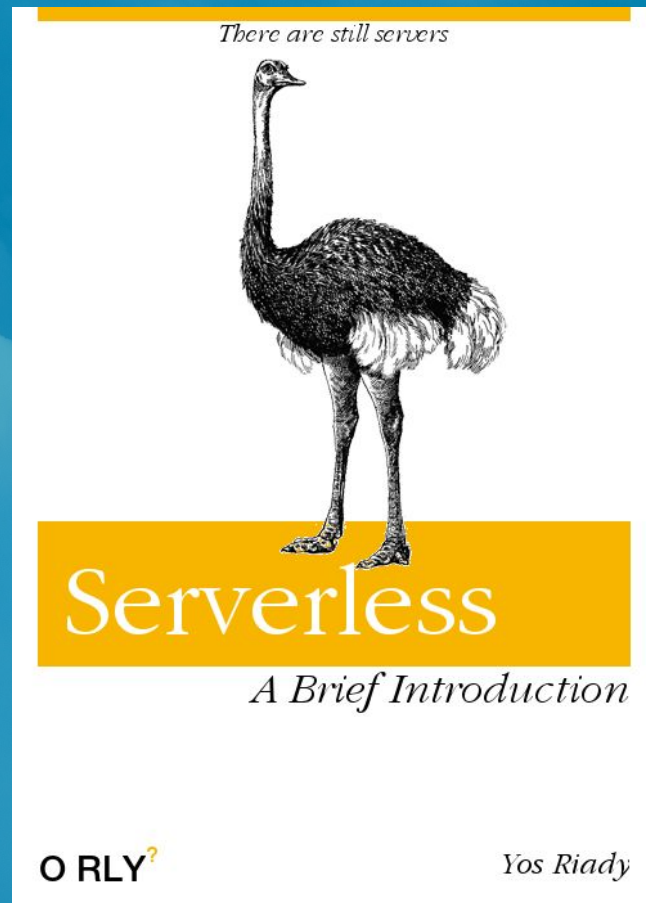


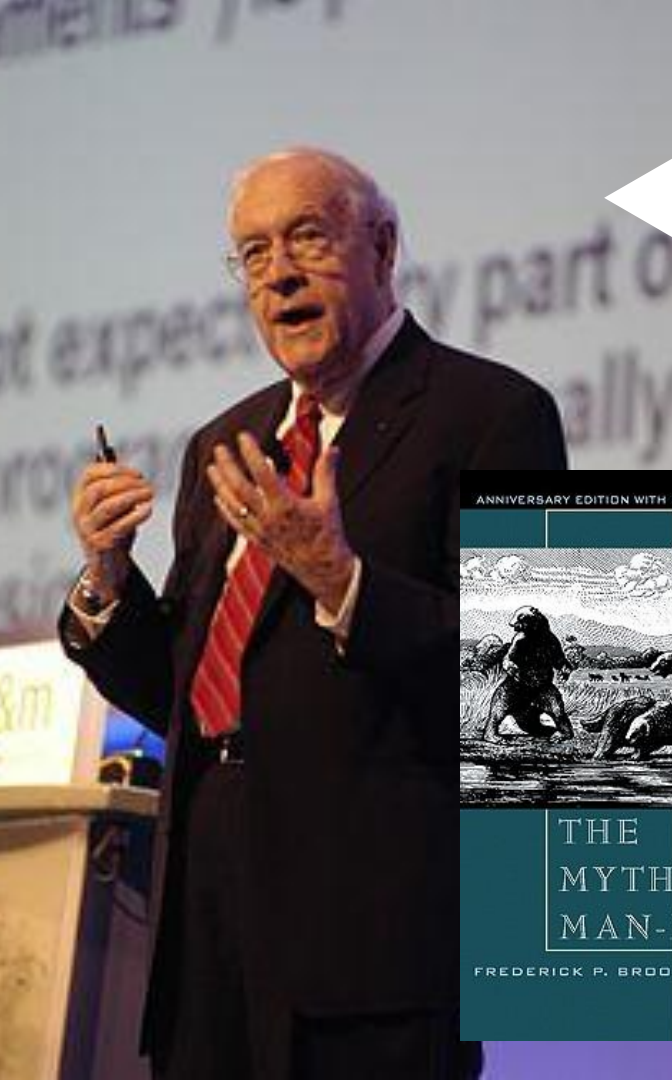
# Serverless

A Brief Introduction

Yos Riady  
yos.io

[bit.ly/2J4wV53](https://bit.ly/2J4wV53)

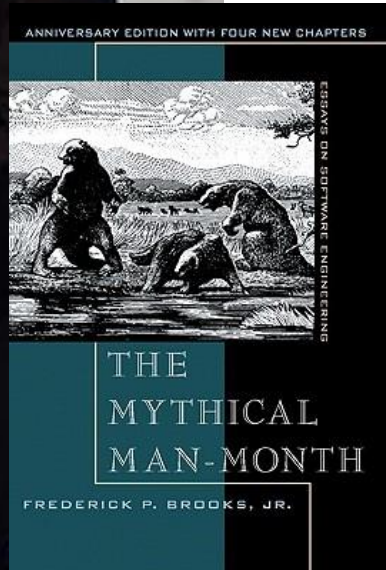




**“The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by the exertion of the imagination.**

Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures....”

from Chapter 1 of Fred Brooks's 1974 *The Mythical Man-Month*.







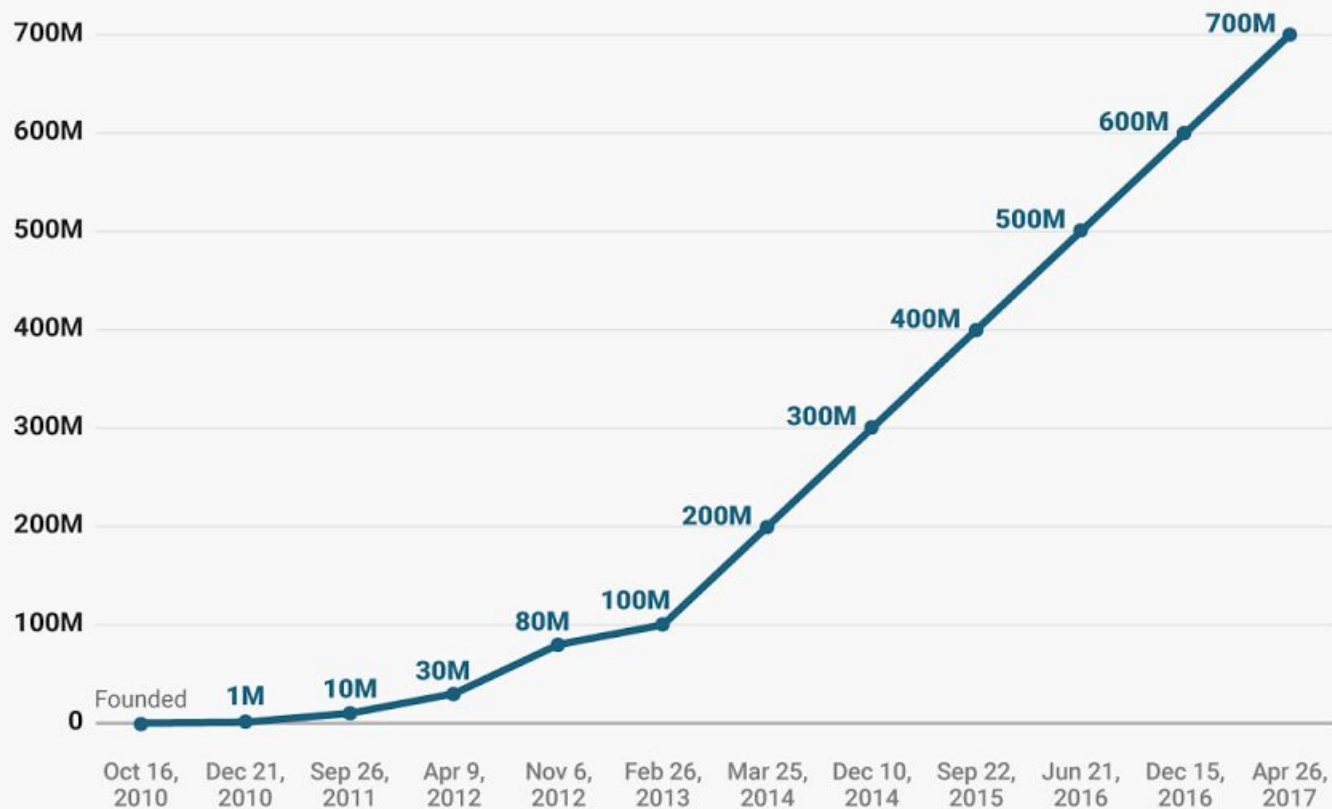


**Your nice software**





# INSTAGRAM MONTHLY ACTIVE USERS, 2010-'17







**Load balancers**











Your software

Load balancers

Service  
registry

Fleet of tires  
containers





Your software

**Load balancers**

*Availability*

**Service  
registry**

*Server  
administration*

*Availability*

*Scalability*

*Security patches*

**Fleet of tires  
containers**

*OS upgrades*



Your software

**Load balancers**

*Availability*

**Service  
registry**

*Server  
administration*

*Availability*

*Scalability*

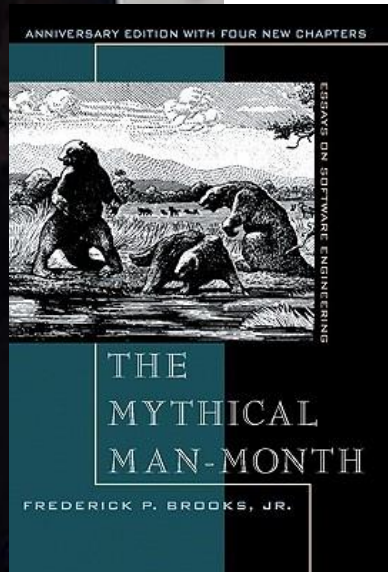
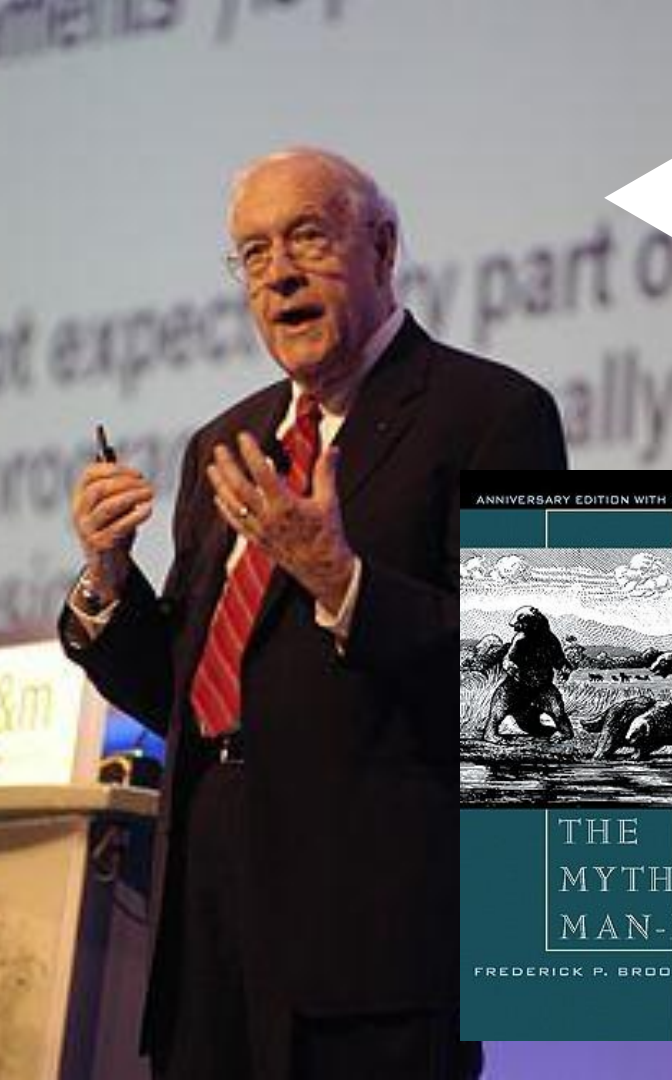
*Security patches*

**Fleet of tires  
containers**

*OS upgrades*







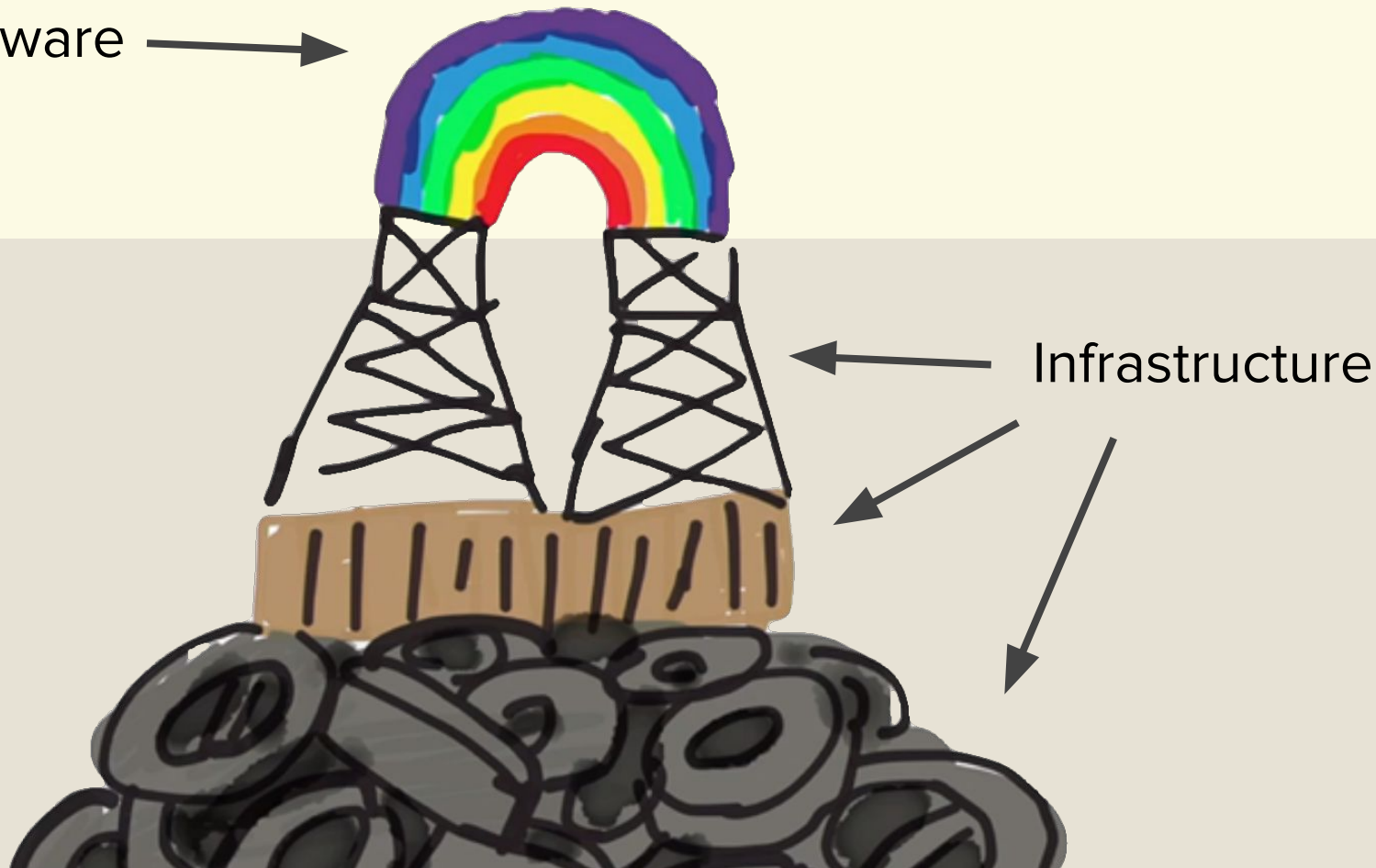
“The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by the exertion of the imagination.

**Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures....”**

from Chapter 1 of Fred Brooks's 1974 *The Mythical Man-Month*.



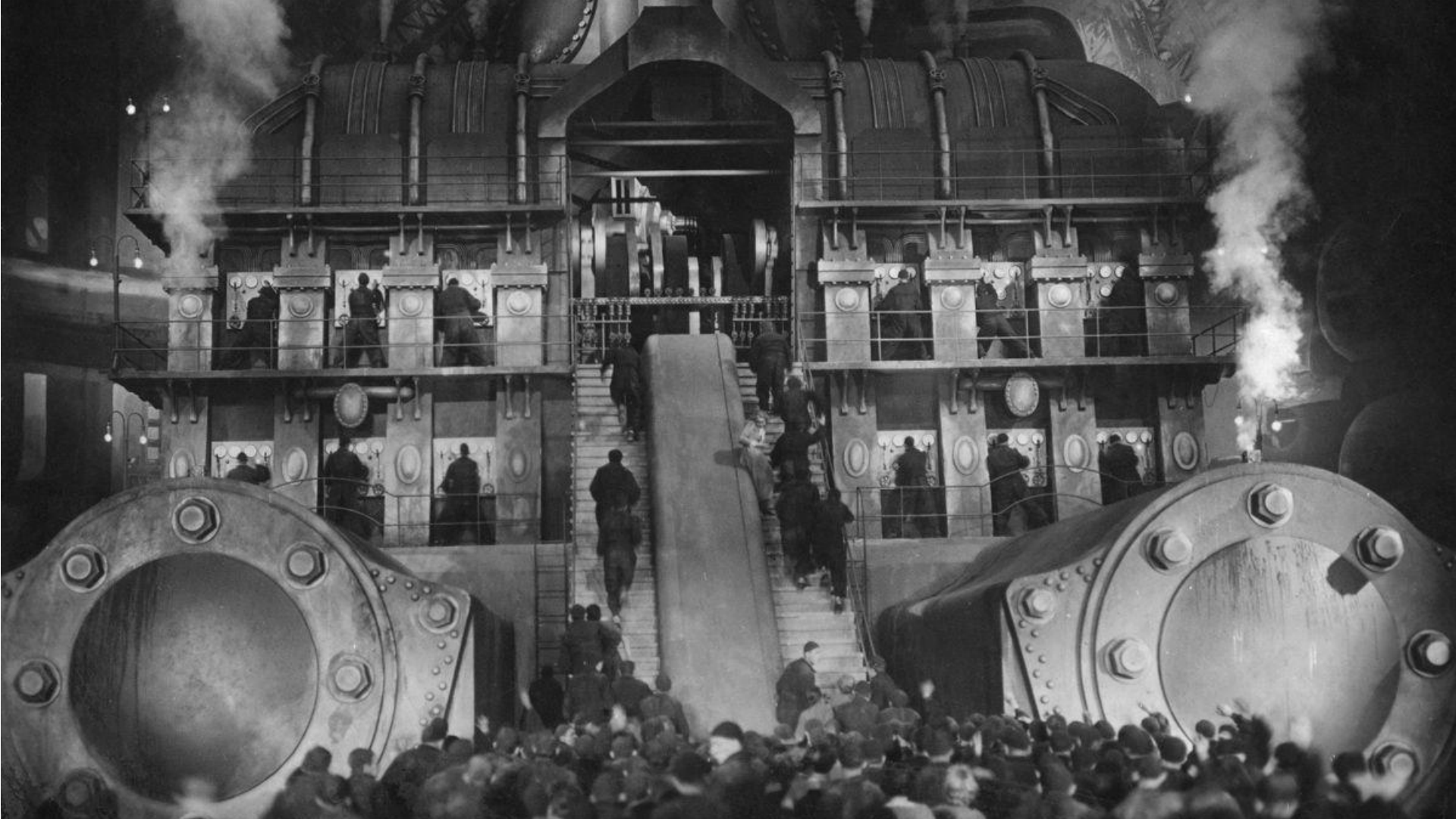
Your software →





**Functional**  
requirements

**Non-functional**  
requirements





1) The host runs Vagrant and Virtualbox

Client  
192.168.2.9

HAProxy Load  
Balancer  
192.168.2.8

2) Controllers control the Kubernetes cluster.  
API calls go there, and state is stored in an etcd cluster

Controller0  
192.168.2.2

Controller1  
192.168.2.3

Controller2  
192.168.2.4

3) The Kubernetes cluster runs services in the 10.32.0.0/16 subnet. iptables nat rules set up by kubernetes transport the packets to the appropriate pod ip in the 10.200/16 subnet (see below)

Worker0  
192.168.2.5

nginx pod  
10.200.0.1

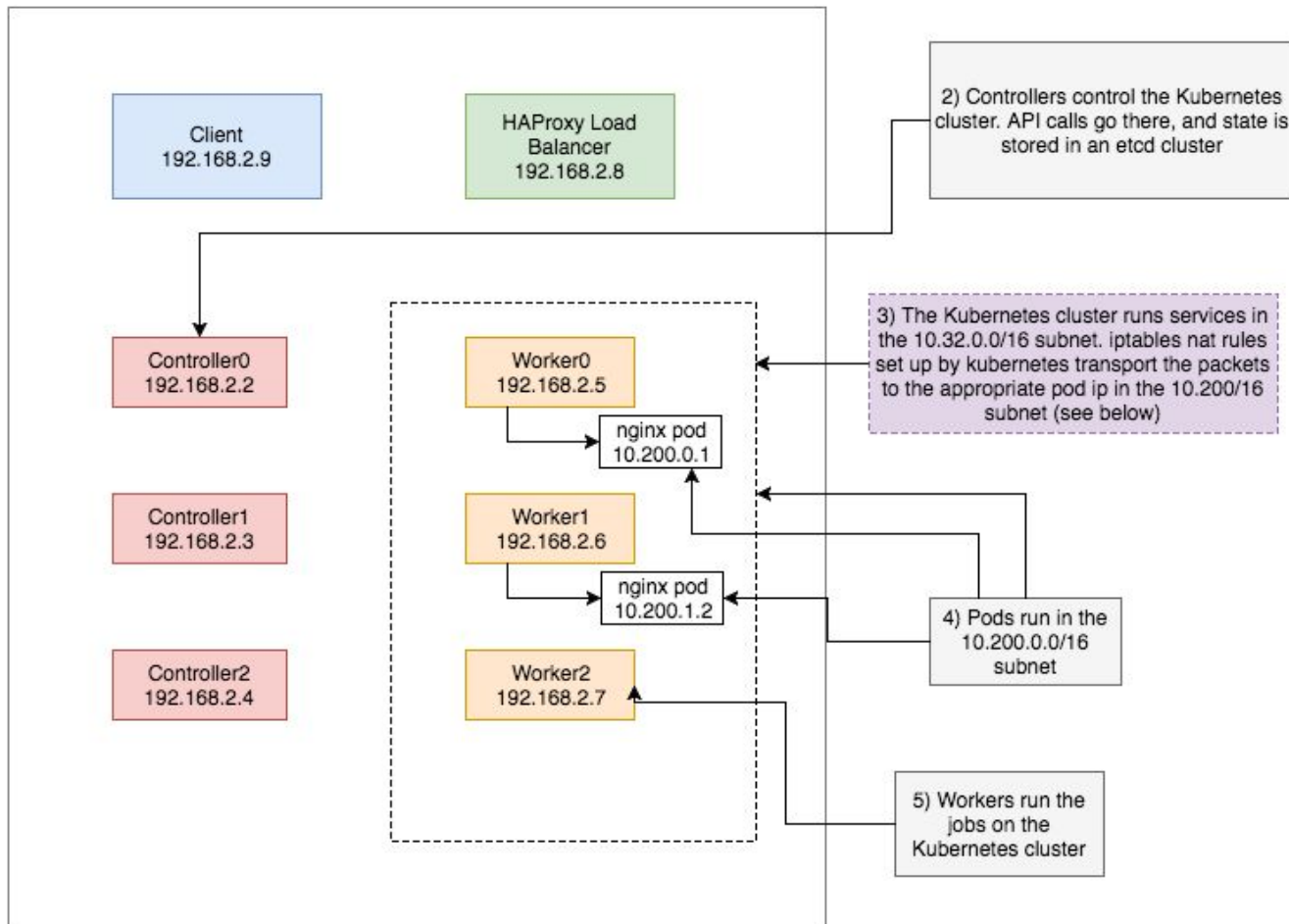
Worker1  
192.168.2.6

nginx pod  
10.200.1.2

Worker2  
192.168.2.7

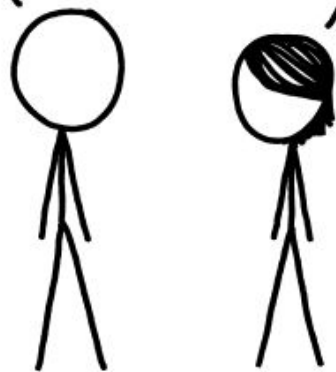
4) Pods run in the 10.200.0.0/16 subnet

5) Workers run the jobs on the Kubernetes cluster

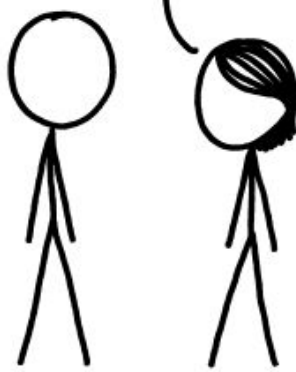


WHAT DO YOU DO?

I MAKE TOOLS  
THAT MAKE TOOLS



...THAT MONITOR CODE  
THAT DEPLOYS TOOLS  
THAT BUILD TOOLS FOR  
DEPLOYING MONITORS...



20 MINUTES LATER...

...FOR MONITORING DEPLOY-  
MENT OF TOOLS FOR—  
BUT WHAT'S IT ALL FOR?)



HONESTLY, NO  
— IDEA.





## **Serverless:**

Letting go of infrastructure concerns to  
focus on functional software

At the end of the day, no one really cares  
about machines, containers, we just care  
about logic and results



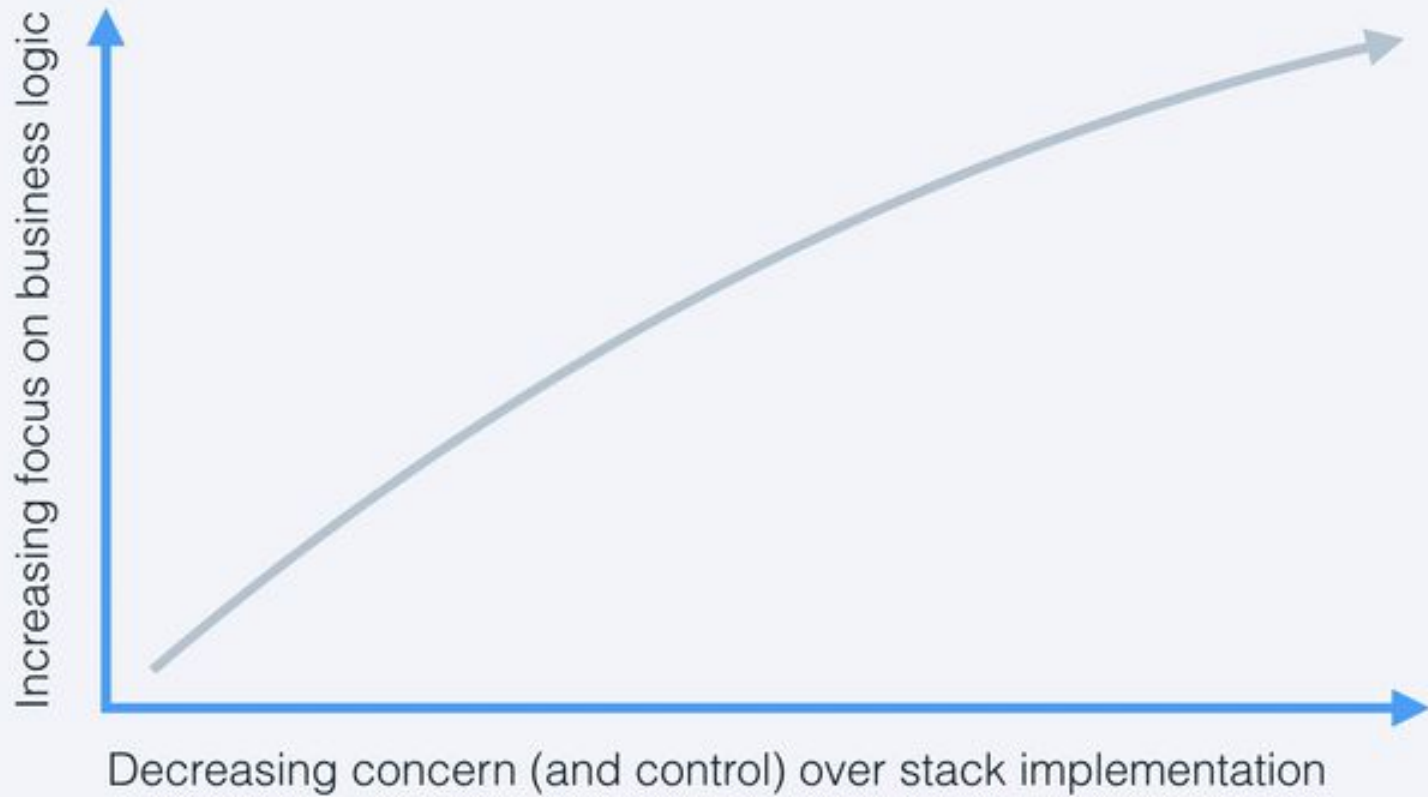
Introducing Apex  
TJ Holowaychuk in Apex Serverless

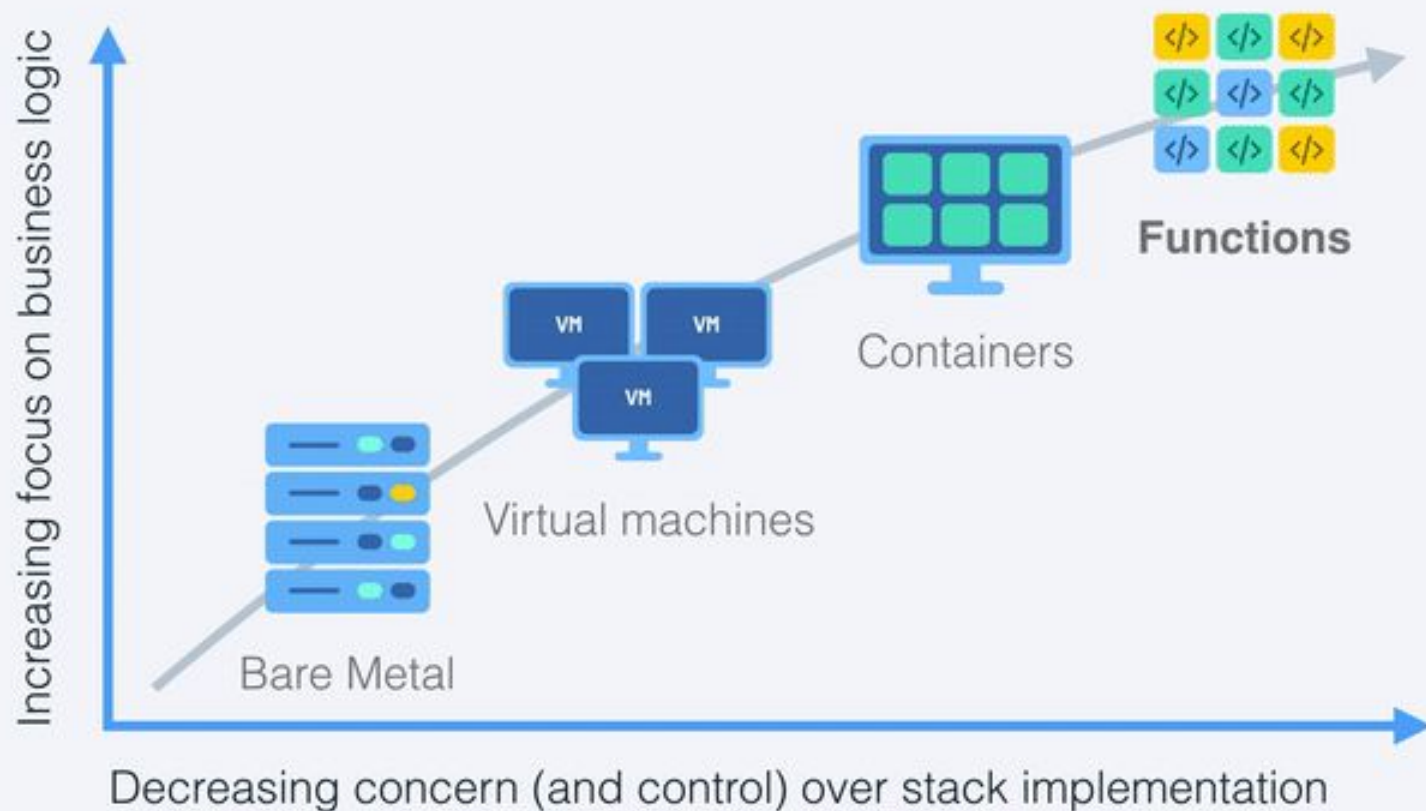


The background of the slide is a solid blue color with a large, soft, white cloud-like shape in the upper left quadrant, creating a sky-like effect.

# Functions-as-a-Service







```
/**  
 * A basic Hello World function  
 * @param {string} name Who you're saying hello to  
 * @returns {string}  
 */  
module.exports = (name = 'world', context, callback) => {  
  callback(null, `hello ${name}`);  
};
```





**AWS Lambda**



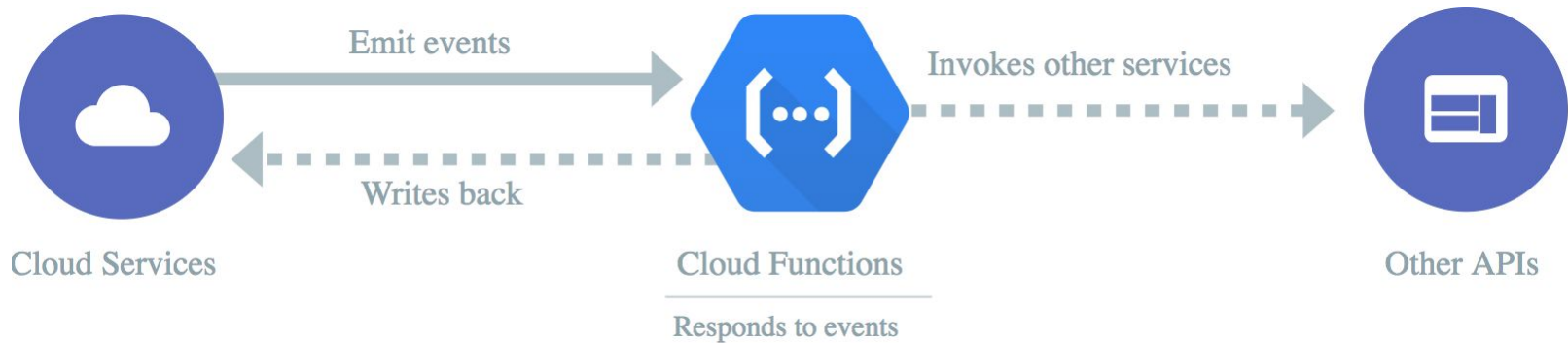
**Google Cloud Functions**



**Azure Functions**



**IBM Cloud Functions**



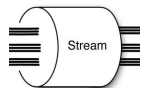
## Events



Changes in data state



Requests to endpoints



Stream messages

Platform events

## Function



Java

Other runtimes

## Services (anything)

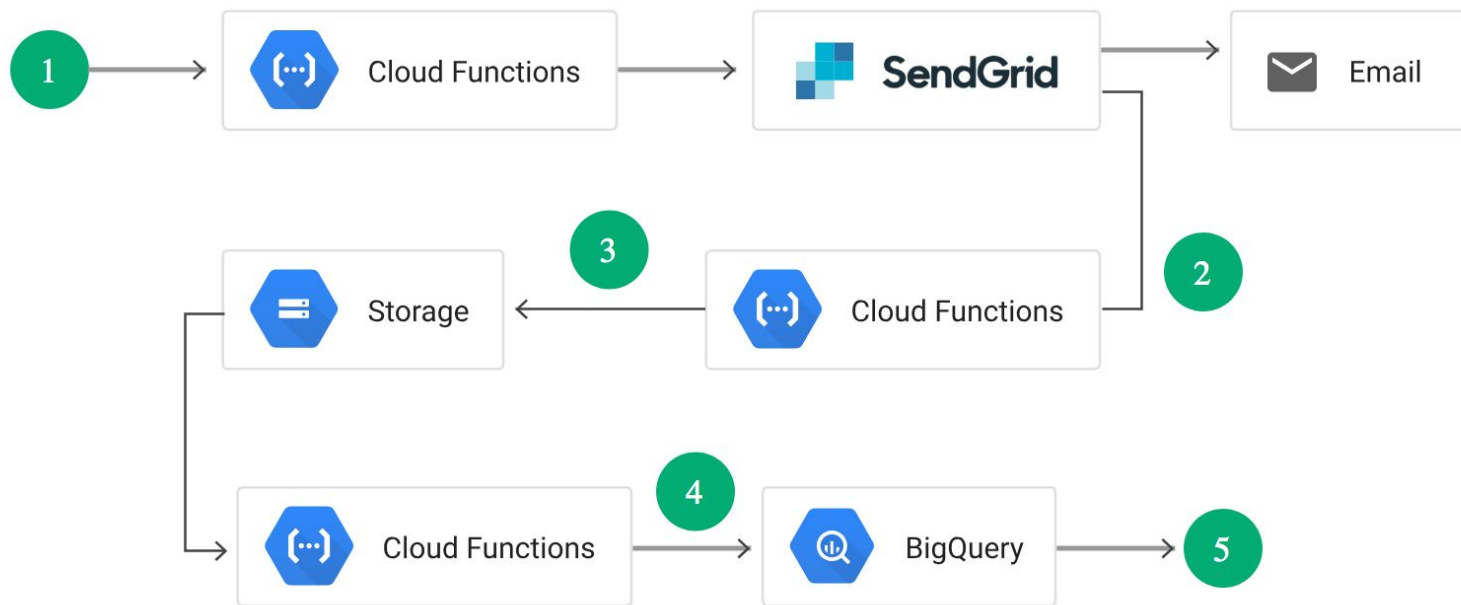


External data / blob stores



HTTP APIs

Other services





The background of the slide is a solid blue color with a large, soft, white cloud graphic on the left side, partially obscuring the blue background.

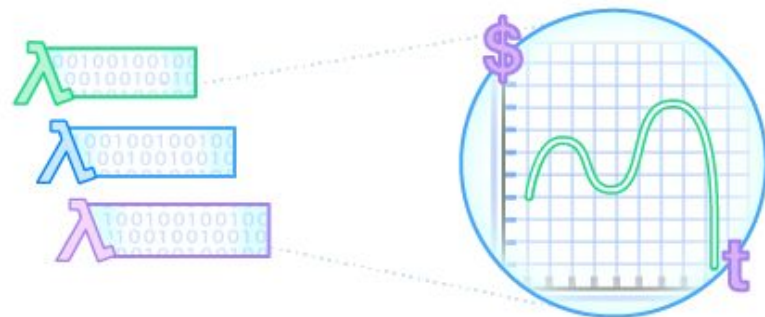
# Perks of Going FaaS



# Less Ops

No provisioning, updating, and managing server infrastructure.

---





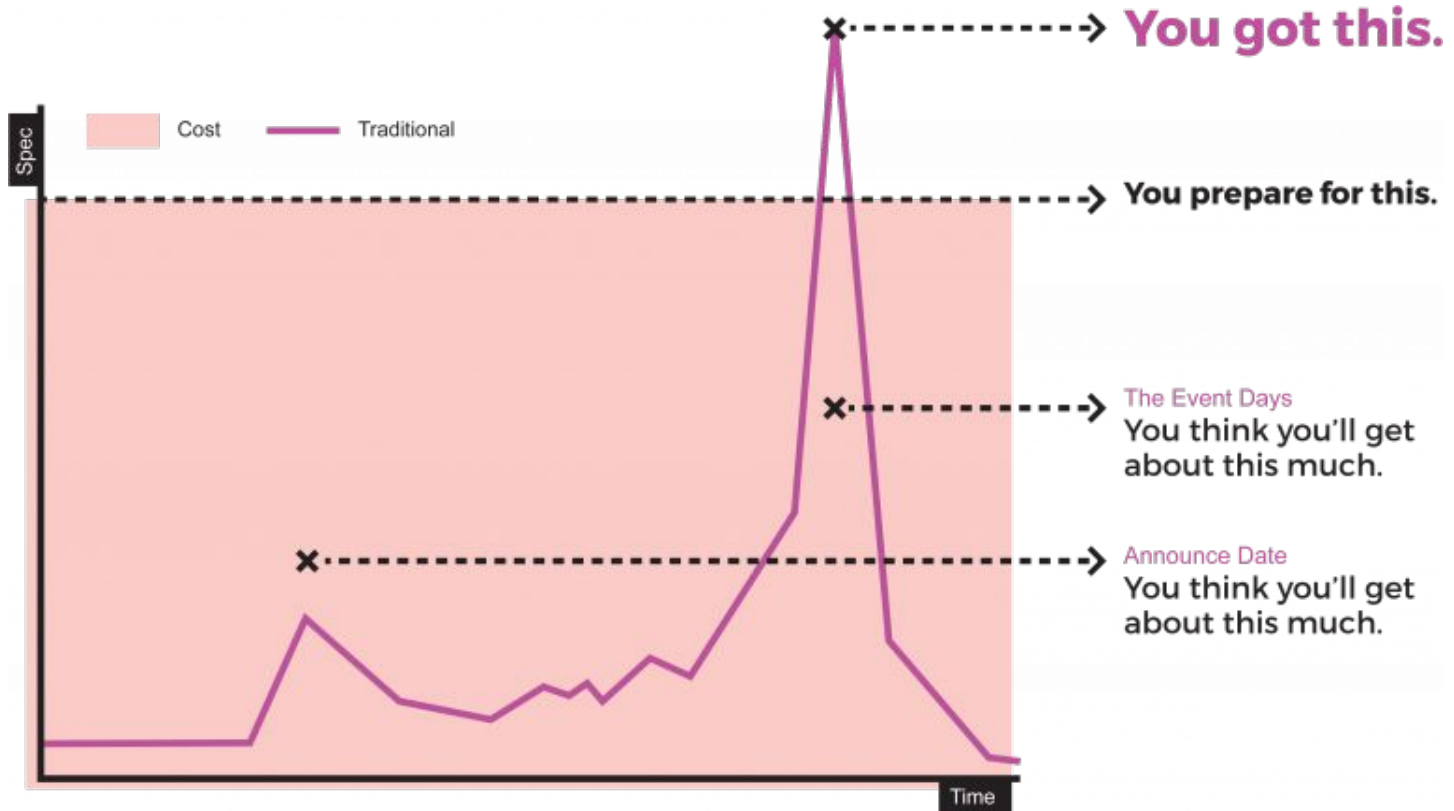


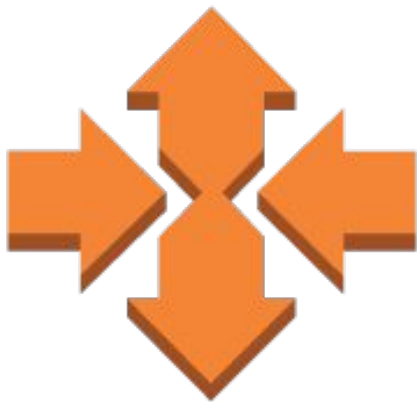
# Pay-per-Execution

No compute cost when idle.

---

# Automatic, Flexible Scaling





# Automatic, Flexible Scaling

No need for capacity planning.

---





# Faster Iteration

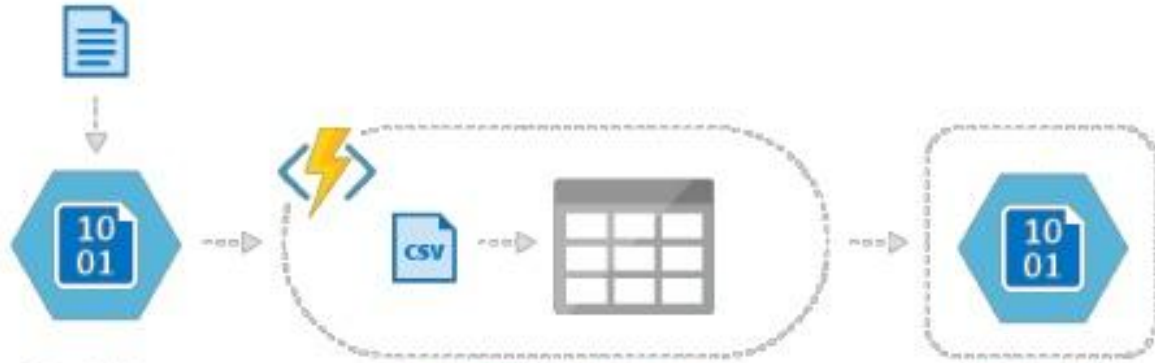
When you have a great idea, the last thing you want to do is set up infrastructure.

---



# FaaS Use Cases

# Multimedia processing



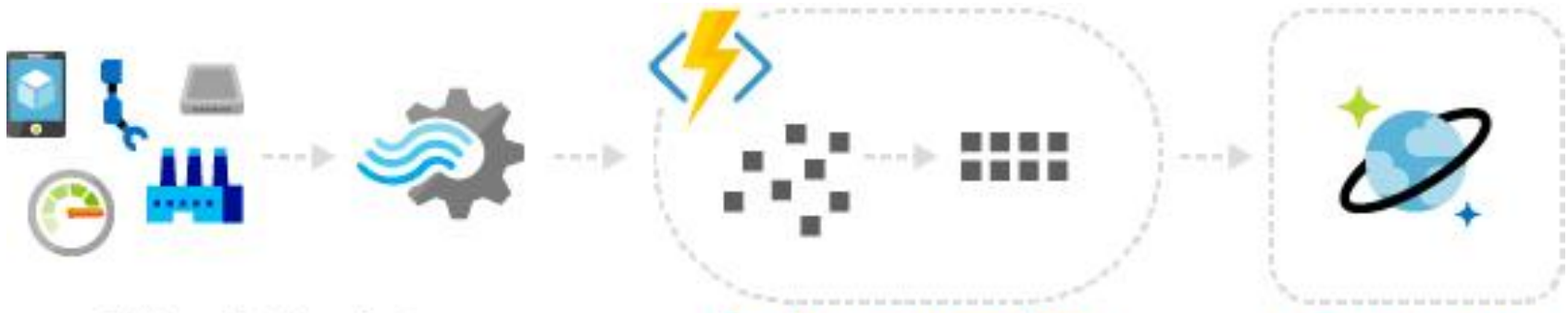
File added to blob storage

Function processes file

Processed file is uploaded



# Stream processing at scale

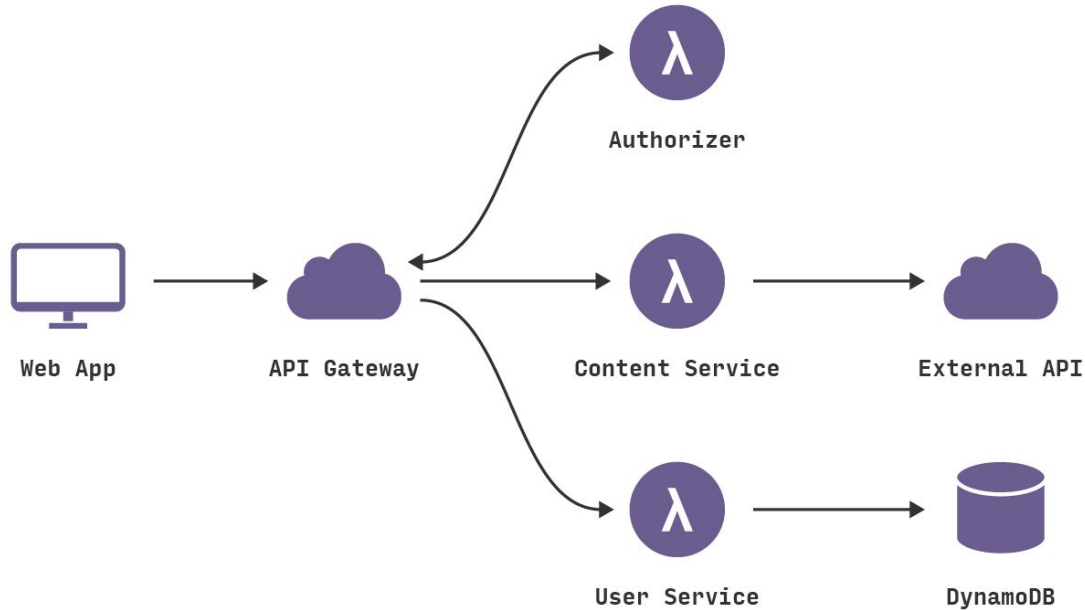


Messages enter the stream

Functions processes message

Store data in a DB

# HTTP REST APIs and Web Applications



# Should you go FaaS?

- Do I **parallelizable, sporadic workloads**?
- Am I creating **event-driven applications**, such as those that respond to database changes, IoT readings, human input etc?
- Do I want to **build applications more quickly** and spend less time wrangling with operational aspects?







**Serverless is still evolving**

# Principles of Serverless Architecture



Use a compute service to execute code on demand



Write single-purpose stateless functions



Design push-based, event-driven pipelines



Create thicker, more powerful front ends



Embrace third party services

**Principles are in flux.**

# Sustaining and Integrating Open Source Technologies

The Cloud Native Computing Foundation builds sustainable ecosystems and fosters a community around a constellation of high-quality projects that orchestrate containers as part of a microservices architecture.



# Standards are in progress.



# cloudevents

A specification for describing event data in a common way

Sign Up

⇌ Why?

Events are everywhere.  
However, event  
publishers tend to  
describe events

📄 What?

Enter CloudEvents, a  
specification for  
describing event data in a  
common way.

👥 Contribute?

This effort is organized via  
the CNCF's [Serverless  
Working Group](#) and  
everyone is encouraged

# Specs are in development.



**NEW** StdLib Launches Code.xyz and Receives \$2M in Funding from Stripe

## Your new API development solution

Standard Library is the fastest way to build, ship, host, document, scale and integrate with APIs without the headache of managing infrastructure and complicated toolchains. See how our serverless platform is changing software development — from internal tools to external developer APIs.

stdlib.com/@username

Sign Up Free

**Receive \$5.00** Secure your namespace now and receive \$5.00 in compute credits

```
> Install command line tools
$ npm install lib.cli -g

> Initialize a workspace
$ lib init

> Create a service
$ lib create

|
|  ● ● ● __main__.js — ~/stdlib/functions
|  |
|  |-> module.exports = async (name = 'Dolores') => {
|      |   return `Hello, ${name}. Bring yourself back online.`;
|      | };
|

> Deploy infinitely scalable web API (dev environment)
$ lib up dev

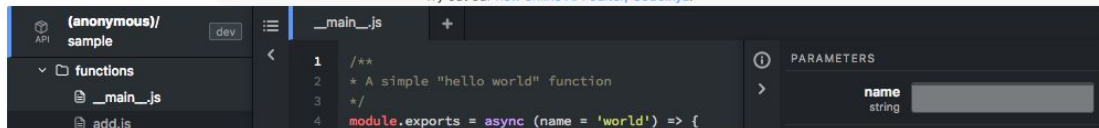
> Deploy a service to production (immutable)
$ lib release
```

build for an ecosystem



**NEWS** Slack · Announces investment in StdLib | TechCrunch · StdLib receives \$2M in Seed Funding

Try out our [new online API editor, Code.xyz!](#)



# New FaaS platforms.

A Brief Introduction to

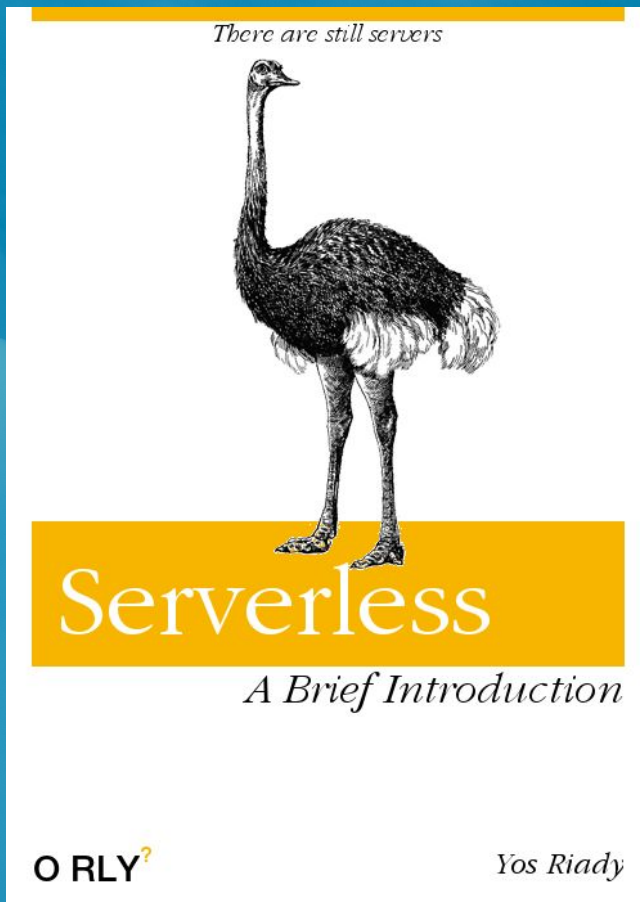
# Serverless



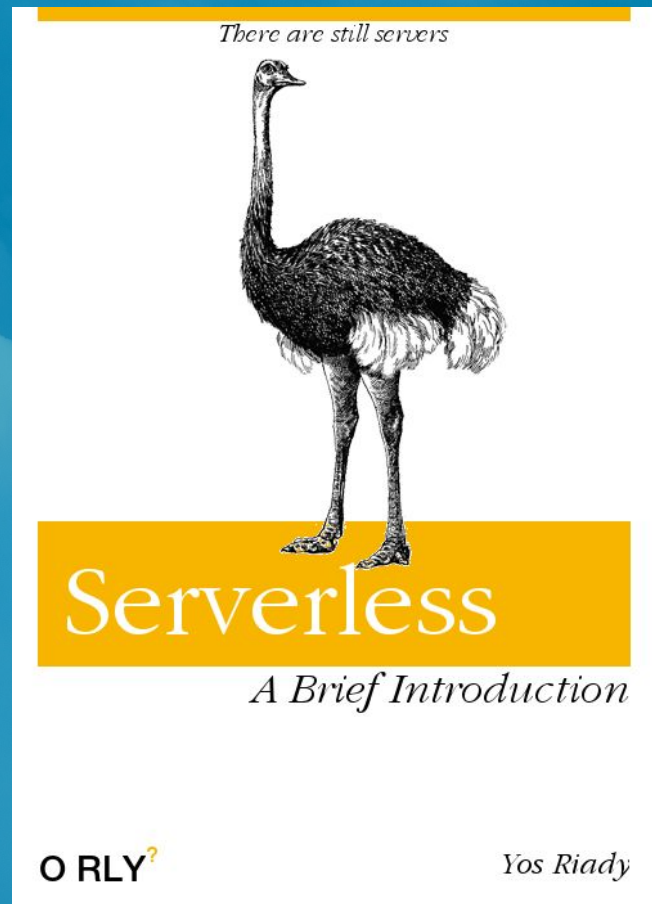
Yos Riady

yos.io

[bit.ly/2J4wV53](https://bit.ly/2J4wV53)



# Thanks







[goingserverless.co](https://goingserverless.co)

WHEN YOU HAVE A GREAT IDEA,

# THE LAST THING YOU WANT TO DO IS SET UP INFRASTRUCTURE.

Going Serverless is a practical guide that teaches you how to design, develop, test, deploy, monitor, and secure Serverless applications from planning to production.

**BUY ON LEANPUB**