

Serverless is dead.

Chris Munns
Principal Developer Advocate
AWS Serverless

A photograph of a brick wall with a faded mural of a person. The text "Why are we here today?" is overlaid in large white letters. In the background, a modern building with a balcony and laundry is visible.

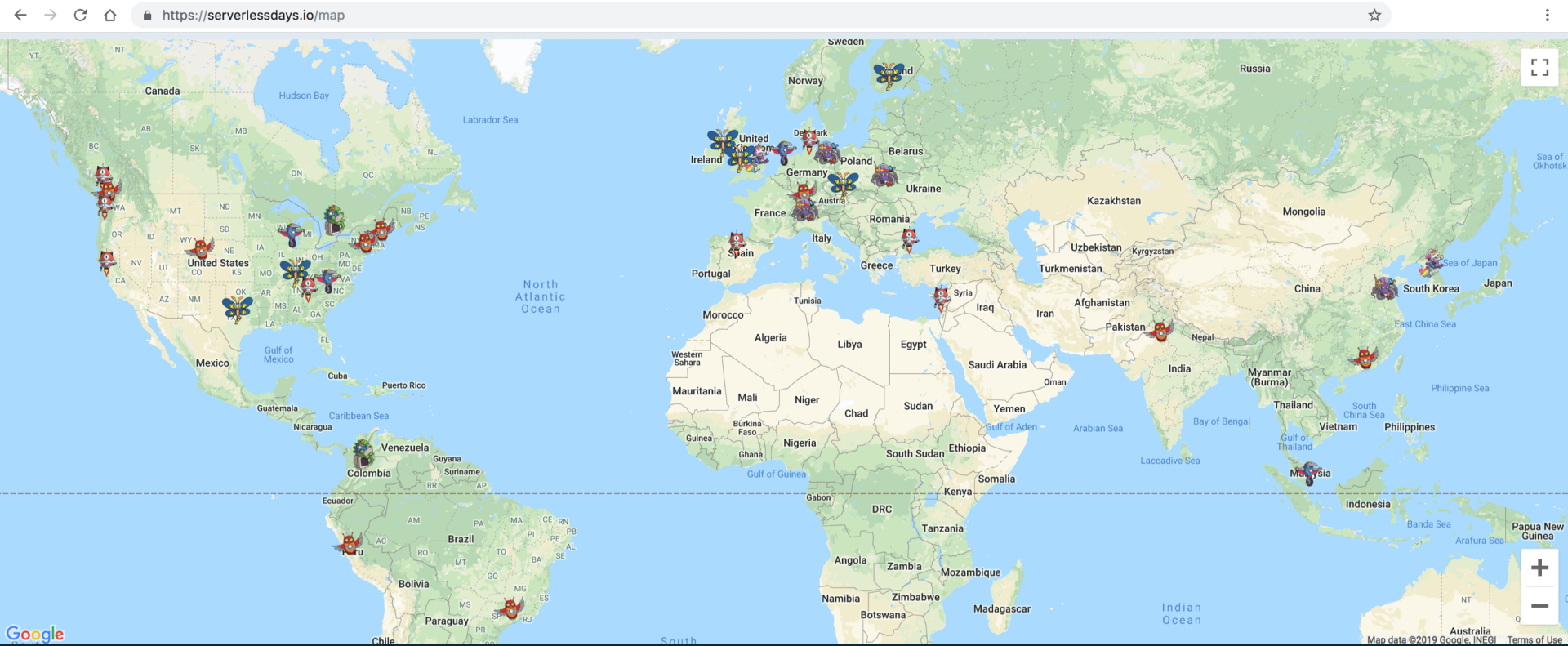
**Why are we
here today?**

A photograph of a cemetery with a cobblestone path and tombstones. The path is made of irregular stones and leads into the distance. On either side of the path are various tombstones and mausoleums, some with classical architectural features like columns and statues. The scene is dimly lit, with trees in the background. The text "Serverless is dead." is overlaid in white on the left side of the image.

Serverless is dead.

Photo by [Echo Grid](#) on [Unsplash](#)

But ServerlessDays?



Serverless is dead.



Photo by [Alexander Andrews](#) on [Unsplash](#)

A close-up photograph of a bulldog lying on its side on a brick-paved surface. The dog has a brindle coat with white markings on its face and chest. It appears to be resting or possibly grieving, with a somber expression. A red leash and a metal chain collar are visible around its neck and front paws. The background is dark and out of focus, showing some industrial or structural elements.

It's ok to grieve

Photo by [Justin Veenema](#) on [Unsplash](#)

What happened?

What happened?

Coroner's report:

What happened?

Coroner's report:

Death by extreme buzzword trauma.

But It's just managed hosting!


But it's just CGI-BIN!

But it's just APIs!

But it's just like PHP!

But it's just like xinetd!

BUT BUT BUT THE SERVERS!



But the security is worse!
But the security is different!
But the cost is higher!
It's too easy!
The name is horrible!
BUT BUT BUT THE SERVERS!



So let us pay our respects

Photo by [Mike Labrum](#) on [Unsplash](#)

By going back to the start

At first there was Amazon S3

At first there was Amazon S3*

At first there was Amazon S3*

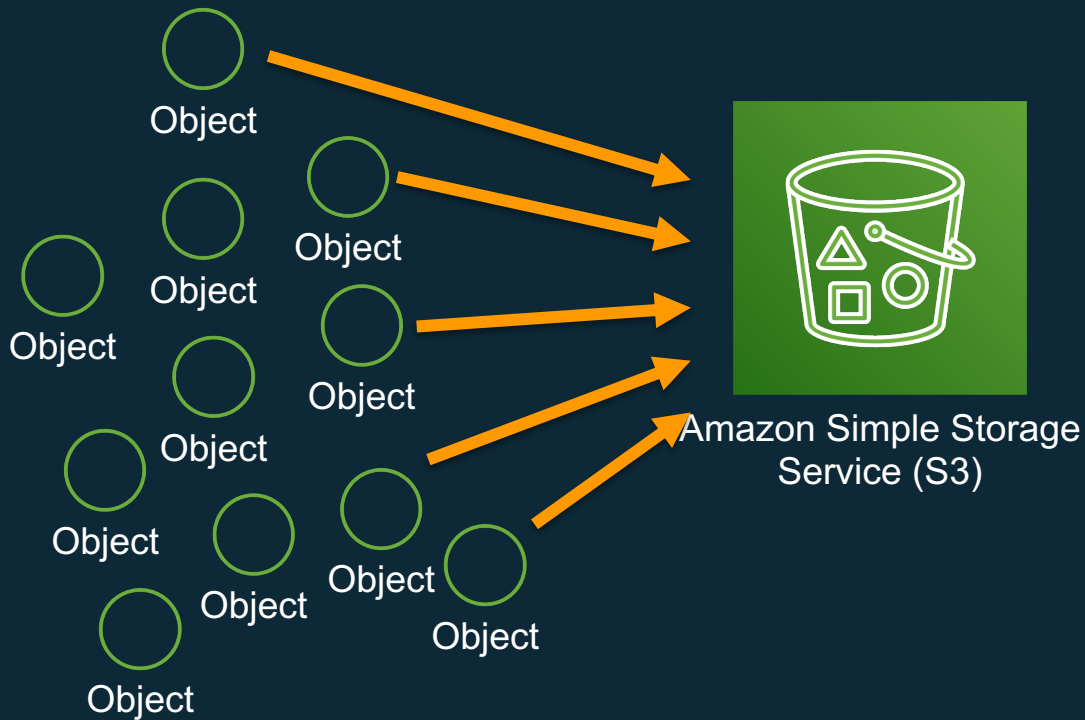
*OK so Amazon SQS was the *first* service announced, S3 was the first to GA.

I have a bucket

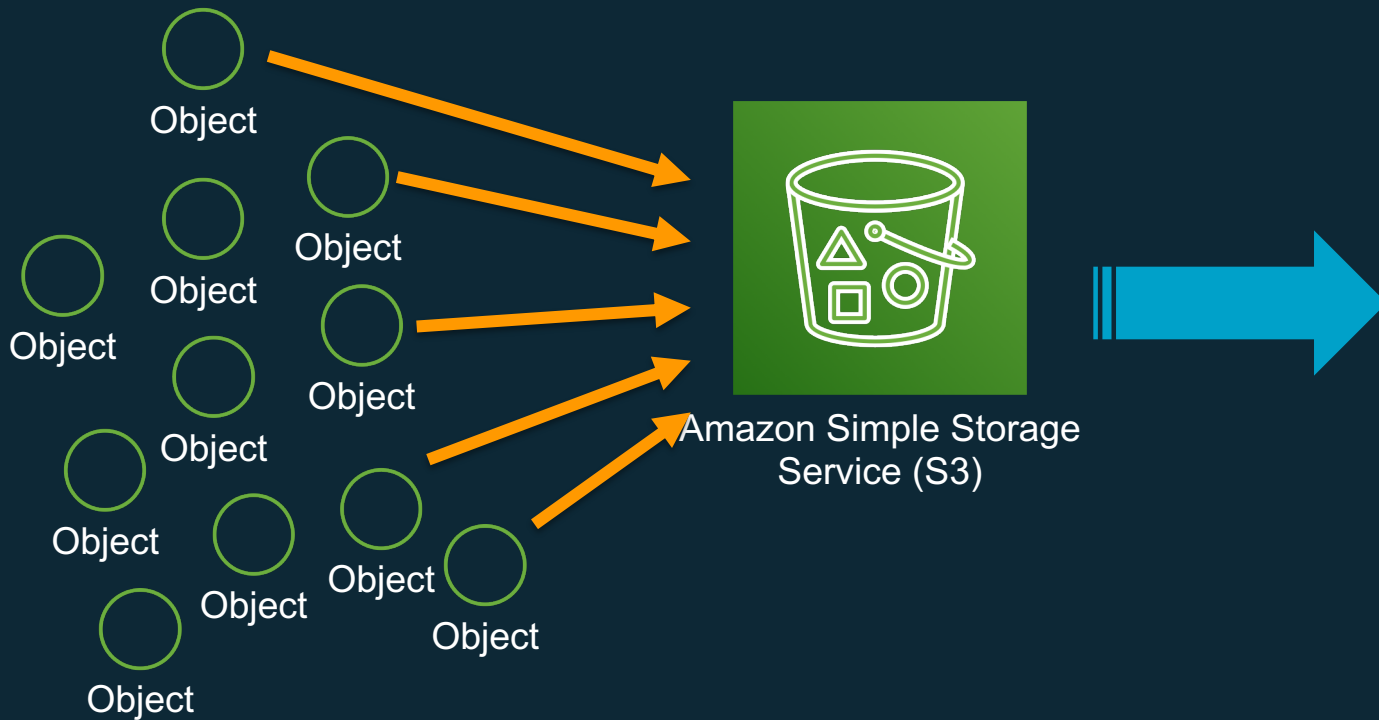


Amazon Simple Storage
Service (S3)

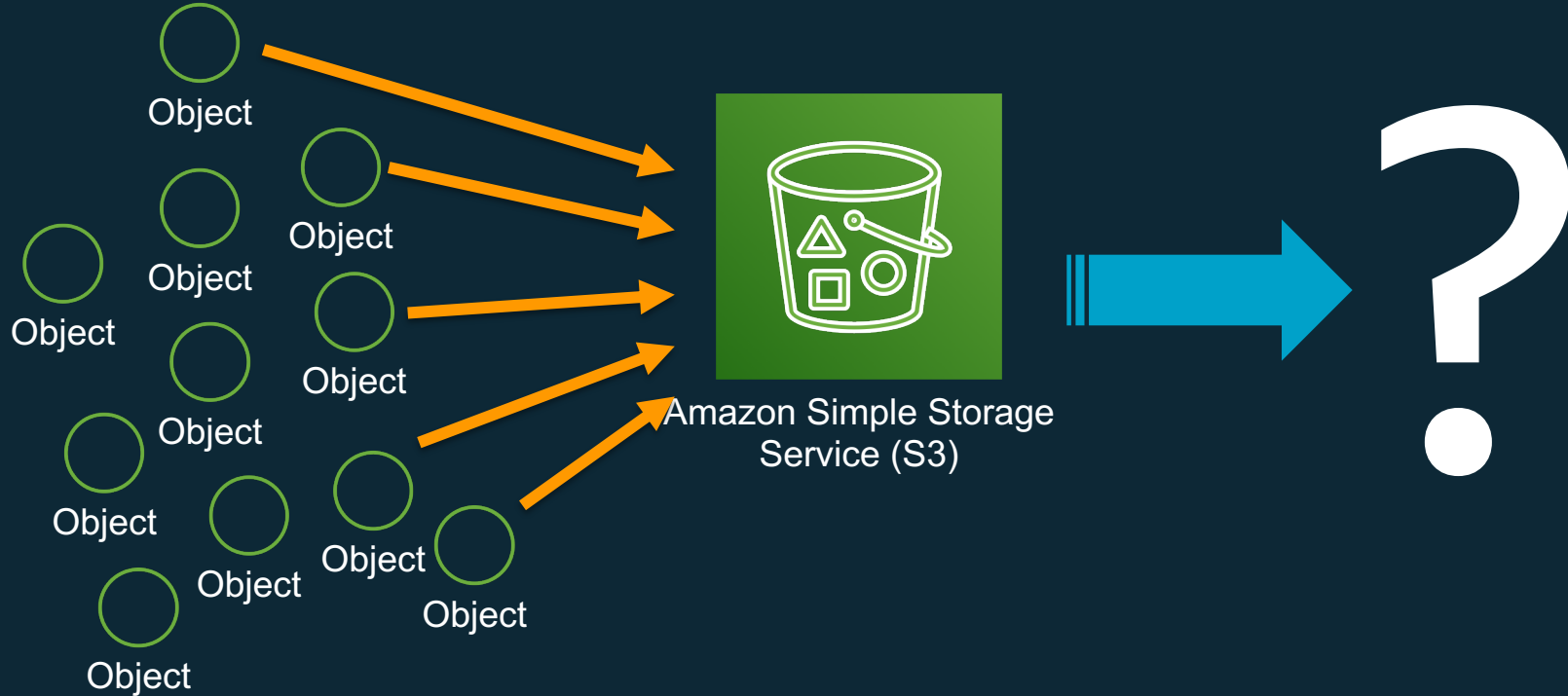
Objects can go in



But then....



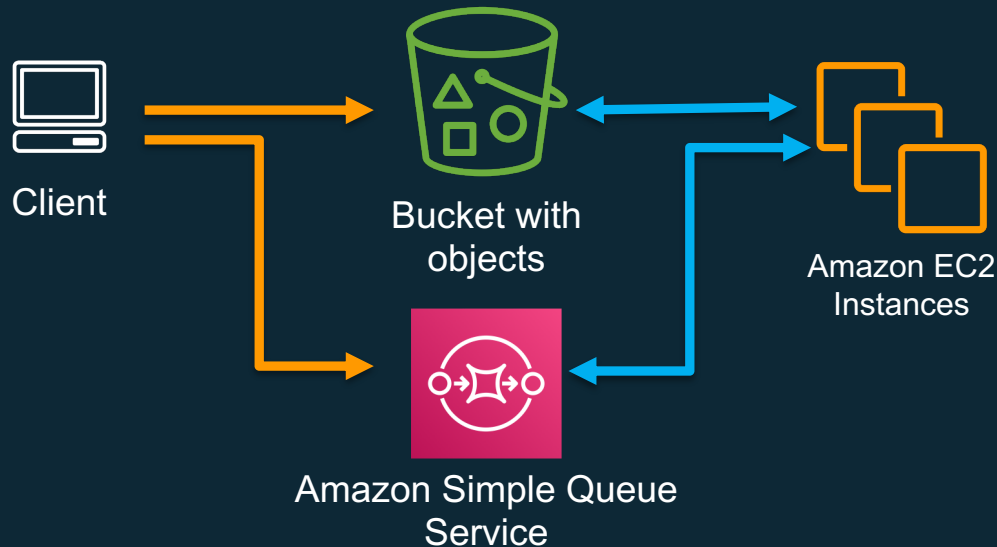
But then what happens from here?



Workflow for S3 stored objects 2006-2014

There were two main ways of working with data in S3 in this time:

- Performing List calls, acting.
- Parallel client calls to "notify" some system that objects were stored, polling that system, acting.



Workflow for S3 stored objects 2006-2014

There were two main ways of working with data in S3 in this time:

- Performing List calls, acting.
- Parallel client calls to "notify" some system that objects were stored, polling that system, acting.



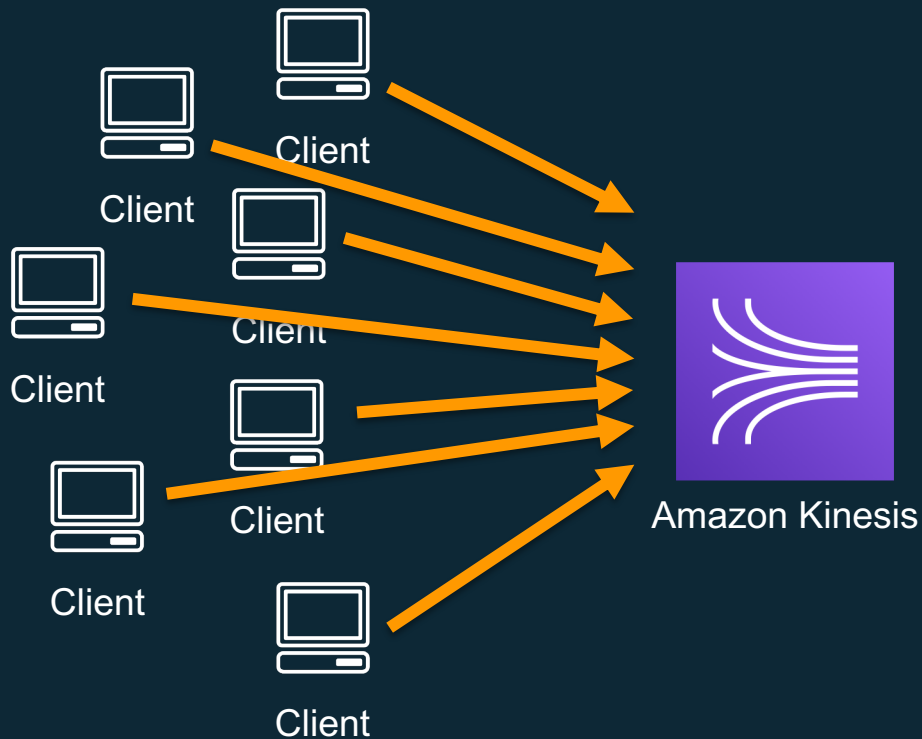
Customers wanted this to be easier



Amazon Simple Storage
Service (S3)

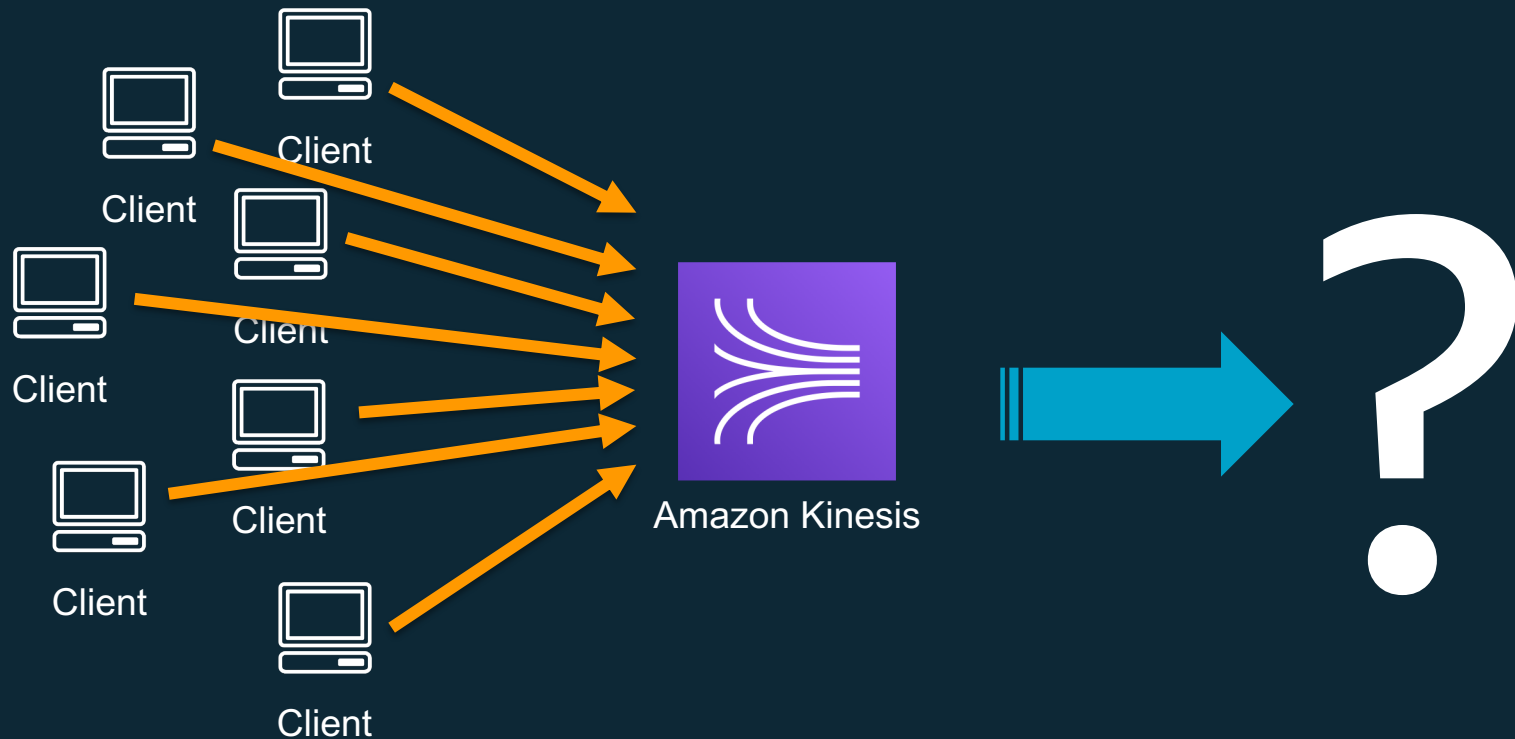


In 2013 AWS Announced Amazon Kinesis



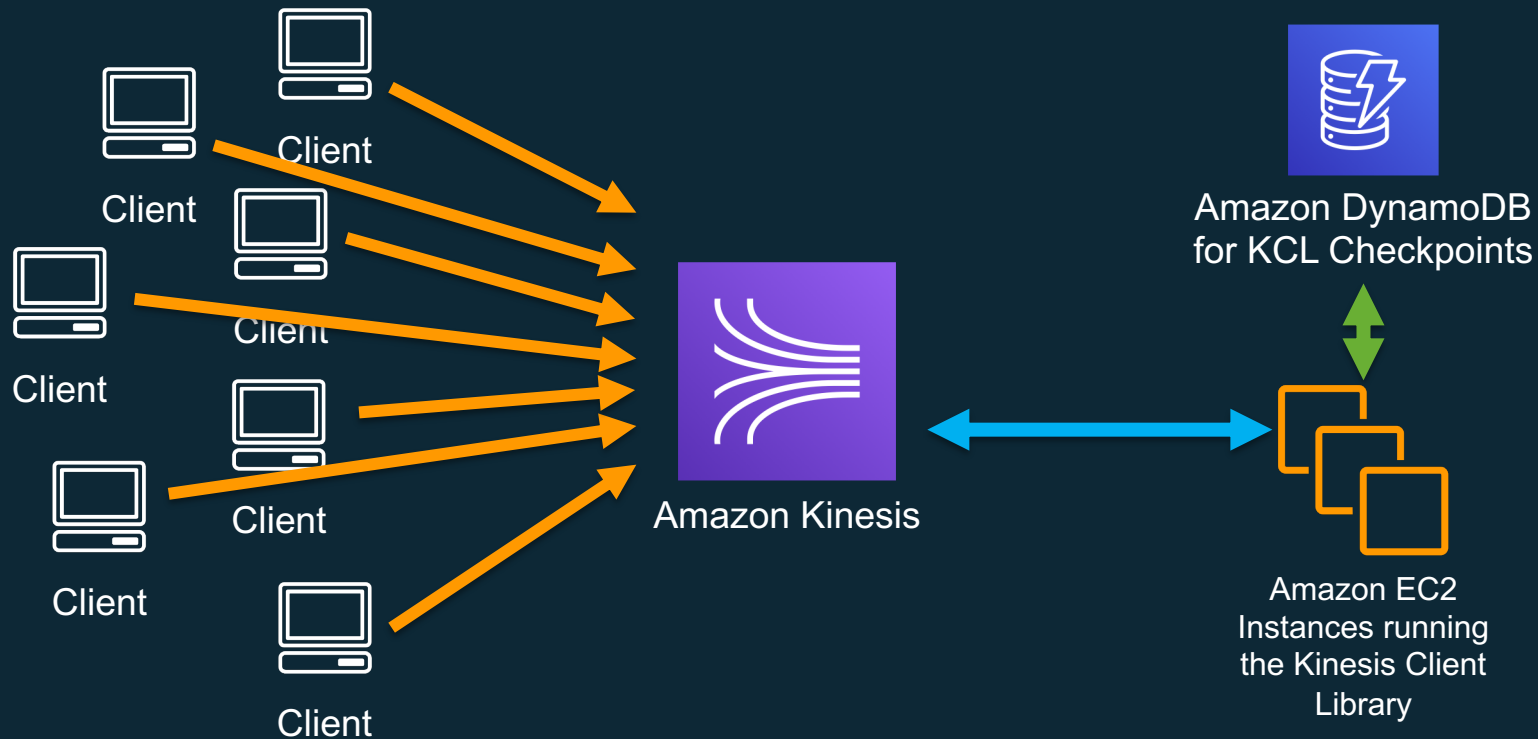
<https://aws.amazon.com/about-aws/whats-new/2013/11/14/introducing-amazon-kinesis/>

In 2013 AWS Announced Amazon Kinesis



<https://aws.amazon.com/about-aws/whats-new/2013/11/14/introducing-amazon-kinesis/>

In 2013 AWS Announced Amazon Kinesis



<https://aws.amazon.com/about-aws/whats-new/2013/11/14/introducing-amazon-kinesis/>

In 2013 AWS Announced Amazon Kinesis



<https://aws.amazon.com/about-aws/whats-new/2013/11/14/introducing-amazon-kinesis/>

And so in November of 2014

All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

The Easiest Way to Compute in the Cloud – AWS Lambda

By Werner Vogels on 13 November 2014 09:30 AM | [Permalink](#) | [Comments \(0\)](#)

When AWS launched, it changed how developers thought about IT services: What used to take weeks or months of purchasing and provisioning turned into minutes with Amazon EC2. Capital-intensive storage solutions became as simple as PUTting and GETting objects in Amazon S3. At AWS we innovate by listening to and learning from our customers, and one of the things we hear from them is that they want it to be even simpler to run code in the cloud and to connect services together easily. Customers want to focus on their unique application logic and business needs – not on the undifferentiated heavy lifting of provisioning and scaling servers, keeping software stacks patched and up to date, handling fleet-wide deployments, or dealing with routine monitoring, logging, and web service front ends. So we challenged ourselves to come up with an easy way to run applications without having to manage the underlying infrastructure and without giving up on the flexibility to run the code that developers wanted. Our answer is a new compute service called [AWS Lambda](#).

[AWS Lambda](#) makes building and delivering applications much easier by giving you a simple interface to upload your Node.js code directly to Lambda, set triggers to run the code (which can come from other AWS services like Amazon S3 or Amazon DynamoDB, to name a couple), and that's it: you're ready to go. AWS handles all the administration of the underlying compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. You can go from code to service in three clicks and then let AWS Lambda take care of the rest.



Contact Info

Werner Vogels
CTO - Amazon.com

werner@allthingsdistributed.com

Other places

Follow werner on [twitter](#) if you want to know what he is current reading or

And so in November of 2014

All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

AWS Lambda makes building and delivering applications much easier by giving you a simple interface to upload your Node.js code directly to Lambda, set triggers to run the code (which can come from other AWS services like Amazon S3 or Amazon DynamoDB, to name a couple), and that's it: you're ready to go. AWS handles all the administration of the underlying compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. You can go from code to service in three clicks and then let AWS Lambda take care of the rest.

is a new compute service called AWS Lambda.

AWS Lambda makes building and delivering applications much easier by giving you a simple interface to upload your Node.js code directly to Lambda, set triggers to run the code (which can come from other AWS services like Amazon S3 or Amazon DynamoDB, to name a couple), and that's it: you're ready to go. AWS handles all the administration of the underlying compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. You can go from code to service in three clicks and then let AWS Lambda take care of the rest.

Contact Info

Werner Vogels
CTO - Amazon.com

werner@allthingsdistributed.com

Other places

Follow werner on [twitter](#) if you want to know what he is current reading or

And so in November of 2014

All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

The Easiest Way to Compute in the Cloud – AWS Lambda

By Werner Vogels on 13 November 2014 09:30 AM | [Permalink](#) | [Comments \(\)](#)

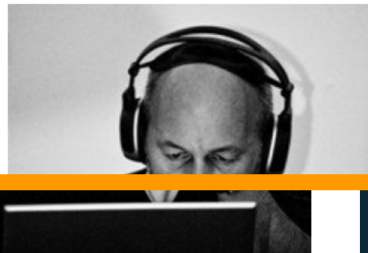
When AWS launched, it changed how developers thought about IT services: What used to take weeks or months of purchasing and provisioning turned into minutes with Amazon EC2. Capital-intensive storage solutions became as simple as PUTting and GETting objects in Amazon S3. At AWS we innovate by

deployments, or dealing with routine monitoring, logging, and web service front ends. So we challenged ourselves to come up with an easy way to run applications without having to manage the underlying infrastructure and without giving up on the flexibility to run the code that developers wanted. Our answer is a new compute service called [AWS Lambda](#).

[AWS Lambda](#) makes building and delivering applications much easier by giving you a simple interface to upload your Node.js code directly to Lambda, set triggers to run the code (which can come from other AWS services like Amazon S3 or Amazon DynamoDB, to name a couple), and that's it: you're ready to go. AWS handles all the administration of the underlying compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging. You can go from code to service in three clicks and then let AWS Lambda take care of the rest.

serverless|

0/0



Contact Info

Werner Vogels
CTO - Amazon.com

werner@allthingsdistributed.com

Other places

Follow werner on [twitter](#) if you want to know what he is current reading or

Well where did #serverless come from?

July 19, 2010. – Techcrunch article about PiCloud

PiCloud Launches Serverless Computing Platform To The Public



Leena Rao @LeenaRao / 9 years ago

 Comment

Startup [PiCloud](#) is launching its platform to the public that allows for batch processing, high performance computing and scientific computing applications. PiCloud, which was incubated at [Lightspeed Ventures](#), claims to “obviate servers,” and enables developers, scientists and engineers to set up applications on the cloud easily.

The PiCloud platform aims to simplify cloud computing through three actions. Cloud.call runs your custom function on the cloud, cloud.result retrieves the return value of your function and cloud.map maps your function to multiple arguments. The startup’s library, which supports the Python code, extracts the byte code, source code, and data necessary to move users’ computation to the cloud. PiCloud also offers a dashboard to manage submitted jobs and analyze usage.

July 19, 2010. – Techcrunch article about PiCloud

PiCloud Launches Serverless Computing Platform To The Public

Leena Rao @LeenaRao / 9 years ago

Comment

Startup PiCloud is bringing its platform to the public. It allows for batch processing, high performance computing and scientific computing applications. PiCloud, which was incubated at Y Combinator, claims to “obviate servers,” and enables developers, scientists and engineers to set up applications on the cloud easily.

The PiCloud platform aims to simplify cloud computing through three actions. Cloud.call runs your custom function on the cloud, cloud.result retrieves the return value of your function and cloud.map maps your function to multiple arguments. The startup’s library, which supports the Python code, extracts the byte code, source code, and data necessary to move users’ computation to the cloud. PiCloud also offers a dashboard to manage submitted jobs and analyze usage.

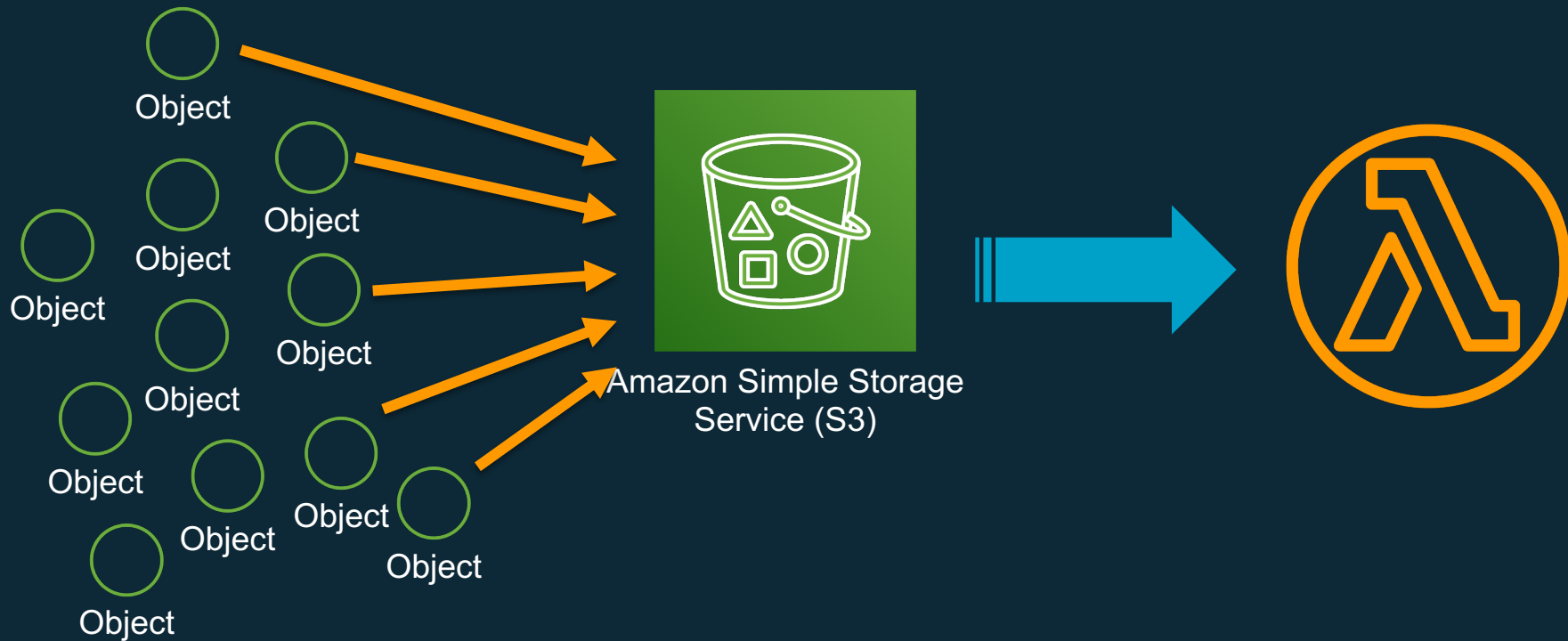
In 2013 Dropbox acquires PiCloud, at that point branded as a “supercomputing” startup.
<https://www.wired.com/2013/11/dropbox-piclou/>

There are no mentions of #serverless

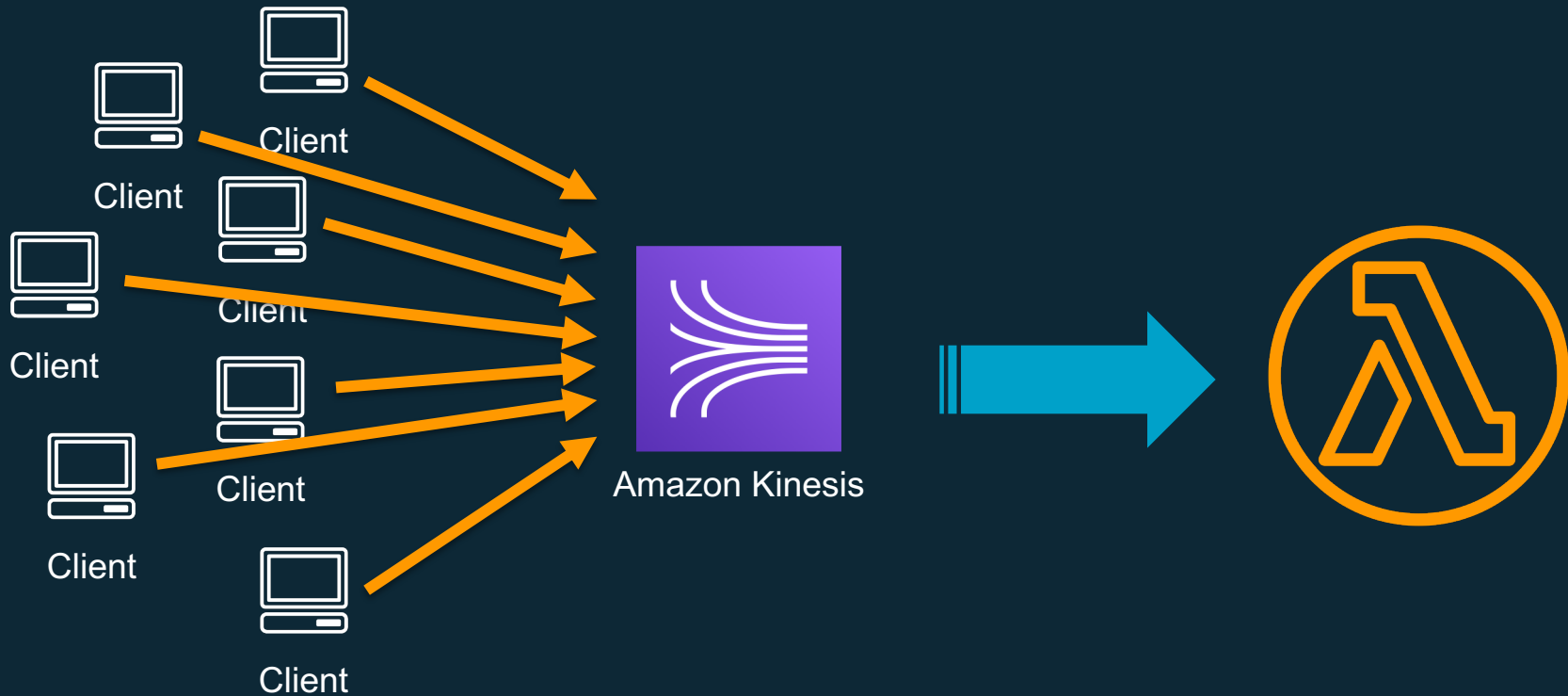
So that brings us back to...



Amazon S3 + AWS Lambda



Amazon Kinesis + AWS Lambda



Alexa, let's change the world.

All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

Amazon announces the Alexa Skills Kit, Enabling Developers to Create New Voice Capabilities

By Werner Vogels on 25 June 2015 10:00 AM | [Permalink](#) | [Comments \(0\)](#)



Contact Info

Werner Vogels
CTO - Amazon.com

werner@allthingsdistributed.com

Other places

Follow werner on [twitter](#) if you want to know what he is current reading or thinking about.

At [werner.ly](#) he posts material that doesn't belong on this blog or on twitter.

Alexa, let's change the world.

All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

The easiest way to build a skill for Alexa is to use [AWS Lambda](#), an innovative compute service that runs a developer's code in response to triggers and automatically manages the compute resources in the AWS Cloud, so there is no need for a developer to provision or continuously run servers. Developers simply upload the code for the new Alexa skill they are creating, and AWS Lambda does the rest, executing the code in response to Alexa voice interactions and automatically managing the compute resources on the developer's behalf.

Using a Lambda function for your service also eliminates some of the complexity around setting up and managing your own endpoint:

- You do not need to administer or manage any of the compute resources for your service.
- You do not need an SSL certificate.
- You do not need to verify that requests are coming from the Alexa service yourself. Access to execute your function is controlled by permissions within AWS instead.
- AWS Lambda runs your code only when you need it and scales with your usage, so there is no need to provision or continuously run servers.
- For most developers, the [Lambda free tier](#) is sufficient for the function supporting an Alexa skill. The first one million requests each month are free. Note that the Lambda free tier does not automatically expire, but is available indefinitely.

Making all of this possible ...

Lambda API



API provided by the Lambda service

Used by all other services that invoke Lambda across all models

Supports sync and async

Can pass any event payload structure you want

Client included in every SDK

Lambda API

API provided by the Lambda service

```
client = boto3.client("lambda")

...

response = client.invoke(
    FunctionName="myFunction"
    InvocationType="Event"|"RequestResponse"
    Payload="json_payload"
)
```

Client included in every SDK

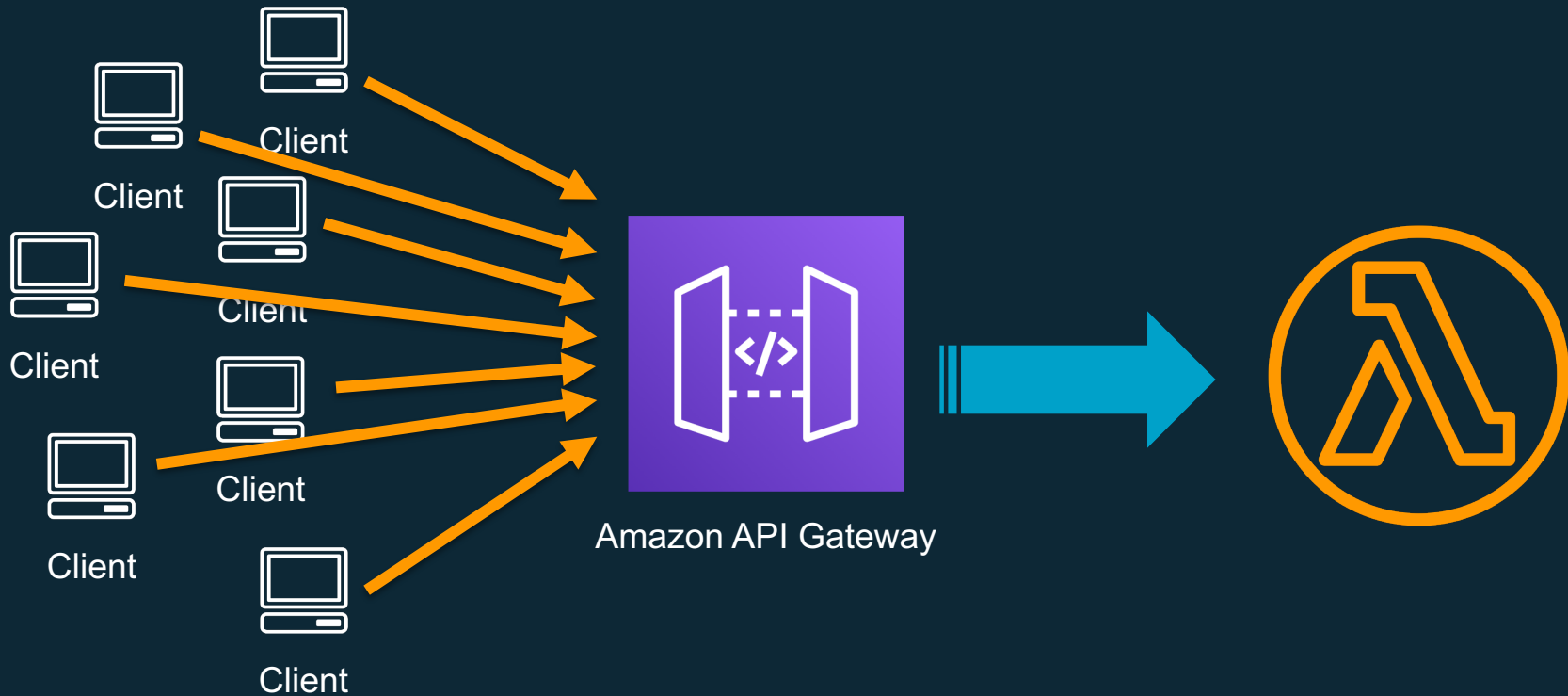
**The Lambda API provides the *model*
for event driven applications the same
way S3's API did for storage.**

And then Amazon API Gateway in July 2015



Amazon API Gateway

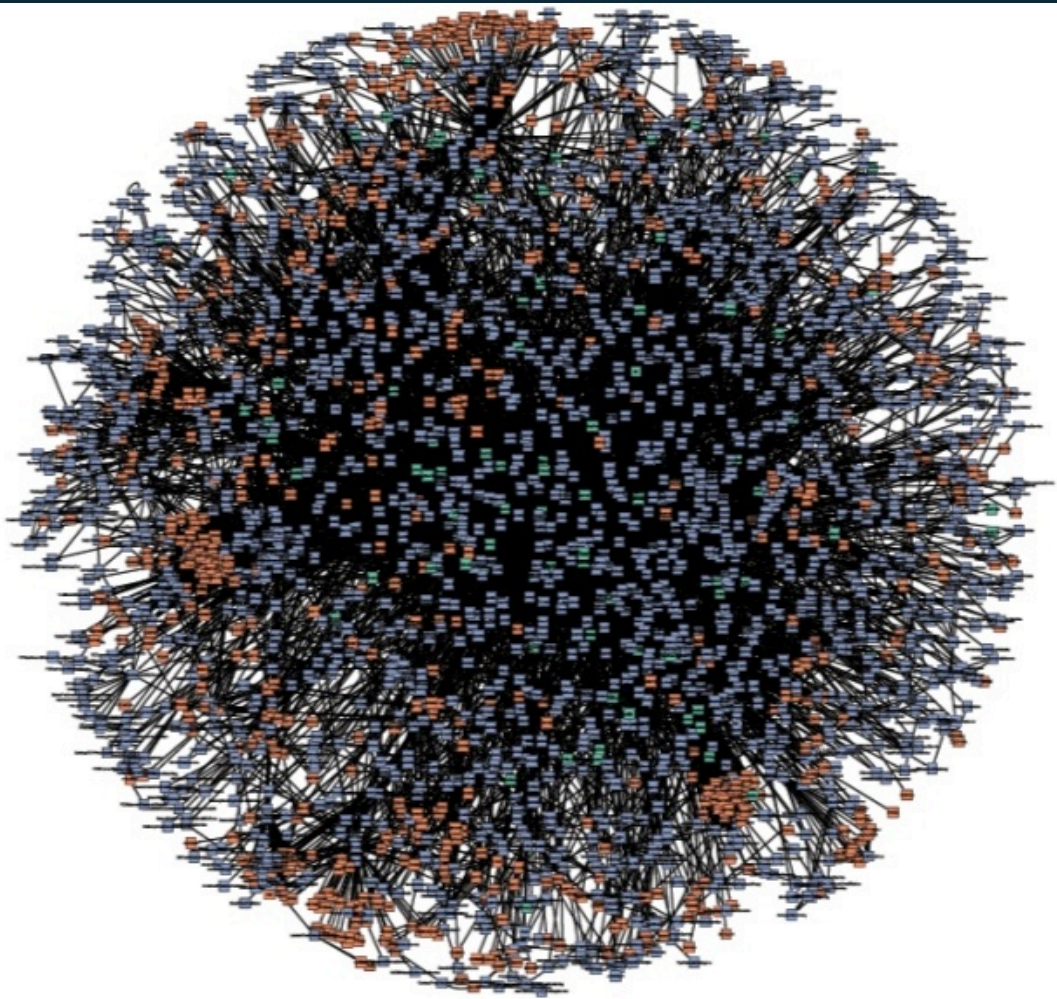
Amazon API Gateway + AWS Lambda



A photograph of a two-story house. The left side of the house has white horizontal siding and red shutters on the windows. The central entrance features a red door with a brass handle and a small window above it, framed by a brown wooden portico. A black metal fence runs along the front of the property. To the right, a portion of a brick house is visible. A red bicycle is parked in the lower right corner. The text "APIs are the front door of microservices" is overlaid in white, bold, sans-serif font across the center of the image.

APIs are the front door of microservices

Photo by [christian koch](#) on [Unsplash](#)



Amazon S3 at launch:



8 separate
microservices



Amazon S3 today:

More than 235
distributed
microservices



APIs power all of these:



iPhone
~11 years



iPad
~9 years



Tesla
Model S
~7 years



Apple
Watch
~5 years



Echo
~5 years



Illumina DNA
Sequencer
~5 years



Amazon
Prime
~14 years



Netflix
Streaming
~12 years



Airbnb
~11 years



U B E R
Uber
~10 years



Square
Square
~10 years



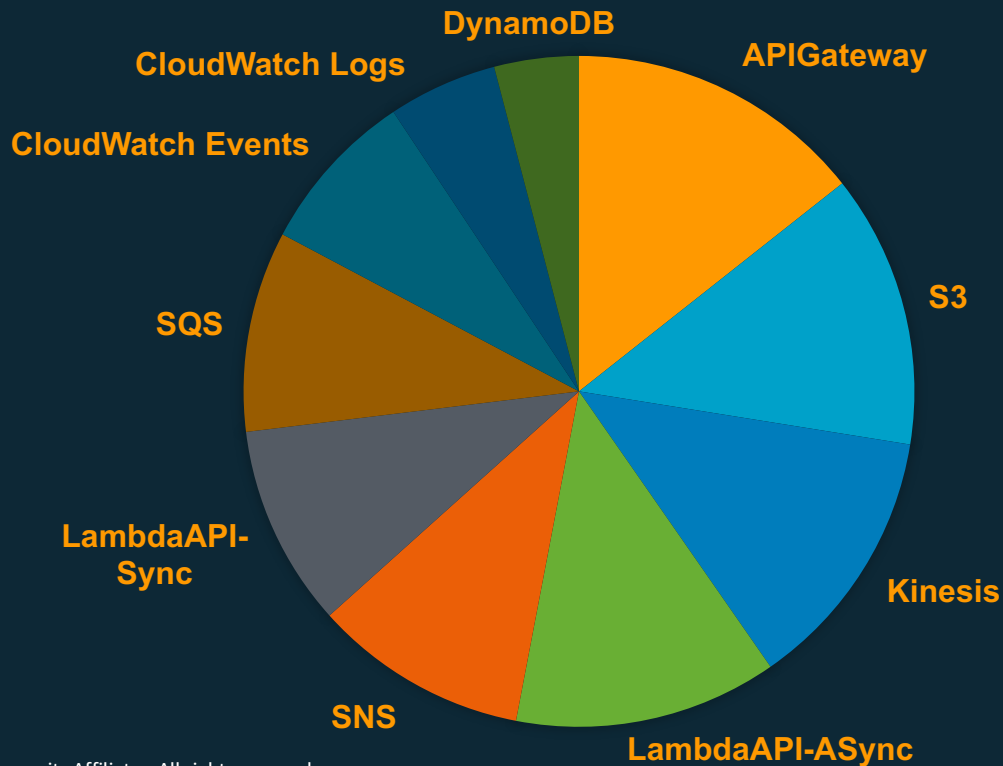
Slack
~6 years

**Some people think that serverless
is just for APIs!**

But it's not.

But it's not.

TOP 10 INVOKE SERVICES BY GB-S, 4 WEEK AVERAGE, RANDOM 4 WEEK SAMPLE





OK, but back to Serverless plz

Photo by [Aaron Burden](#) on [Unsplash](#)

Serverless == FaaS?

At some point Serverless start just being equated with FaaS.

- No AWS announcement in 2015 called Lambda or API Gateway "Serverless"
- AWS re:Invent 2015 featured talks with the word serverless in the title specific to Lambda
- Nov 2015 Techcrunch: "AWS Lambda Makes Serverless Applications A Reality"
- Feb 2016 IBM announces OpenWhisk, an open source serverless FaaS
- May 2016 - first Serverless Conf in NYC, ~13 months after the GA of AWS Lambda

Serverless == Faas?

And then the space grew

- Nov 2016 – Azure Functions GA
- Dec 2016 – OpenFaaS launch
- Oct 2017 – Oracle FN Project release
- Aug 2018 – Google Cloud Functions GA

Serverless abuse

But then, everything became serverless..

- managed container offerings
- PaaS-like services
- storage services
- messaging services
- serverless the flamethrower! (Spaceballs joke)
- etc, etc, etc

By late 2017/early 2018 the term started getting pretty stretched

A knight in full plate armor is riding a white horse. The knight is holding a sword aloft in his right hand. The horse is wearing a surcoat with blue and orange horizontal stripes. The background is a blurred landscape with trees and a cloudy sky.

Some of us fought in vain to keep
the term “pure”

Photo by [Andrew Yardley](#) on [Unsplash](#)



**But our enemies
were numerous..**

Photo by [Ray Hennessy](#) on [Unsplash](#)

So does the word Serverless even matter?

Is Serverless really just a spectrum?

In a November 1st 2017 post on the A Cloud Guru blog, AWS Community Hero Ben Kehoe wrote about the “spectrum” of serverless:

- **Service-full + ephemeral compute**
- **Tighter correspondence between resources used and resources billed**
- **Smaller and more abstracted control plane**

<https://bit.ly/2BBy4L5>



@ben11kehoe

The concept of the spectrum made a lot of sense



Self managed FaaS



AWS Fargate



AWS Lambda



AWS AppSync

The concept of the spectrum made a lot of sense



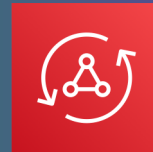
Self managed FaaS



AWS Fargate



AWS Lambda



AWS AppSync

You can measure across: cost, complexity, operational burden, flexibility, portability, etc

The concept of the spectrum made a lot of sense

Biased opinion is that gravity is dragging the industry
this way →



Self-managed FaaS



AWS Fargate



AWS Lambda



AWS AppSync

You can measure across: cost, complexity, operational burden, flexibility, portability, etc

Then the folks from UC Berkeley weighed in (again)

Cloud Programming Simplified: A Berkeley View on Serverless Computing



*Eric Jonas
Johann Schleier-Smith
Vikram Sreekanti
Chia-Che Tsai
Anurag Khandelwal
Qifan Pu
Vaishaal Shankar*

Then the folks from UC Berkeley weighed in (again)

Cloud Programming Simplified: A Berkeley View on Serverless Computing

We predict that serverless use will skyrocket. We also project that hybrid cloud on-premises applications will dwindle over time, though some deployments might persist due to regulatory constraints and data governance rules.

21

- While serverful cloud computing won't disappear, the relative importance of that portion of the cloud will decline as serverless computing overcomes its current limitations.
- Serverless computing will become the default computing paradigm of the Cloud Era, largely replacing serverful computing and thereby bringing closure to the Client-Server Era.

Qifan Pu
Vaishaal Shankar

So does the word Serverless even matter?

no.

Serverless is Dead

What we should instead be focusing on is what we're seeing to be the new way of doing modern application development

- Greatly reduced operational burden
- Tighter alignment to costs w/ usage
- Developers can/could/should be able to do almost anything
 - understated warning to Ops/DevOps/SRE/-ish folks
- Opinionated platforms that allow for multiple use-cases
- Opinionated platforms that bake in true best of breed practices, security, scale, performance, cost aspects for you

What matters:

What matters:

Greater agility

Less overhead

Better focus

Increased scale

More flexibility

Pay for value

A promise to ourselves

A promise to ourselves

We'll focus on the problems and solutions and not the buzzword!

Don't feed the trolls!

Ignore the FUD!

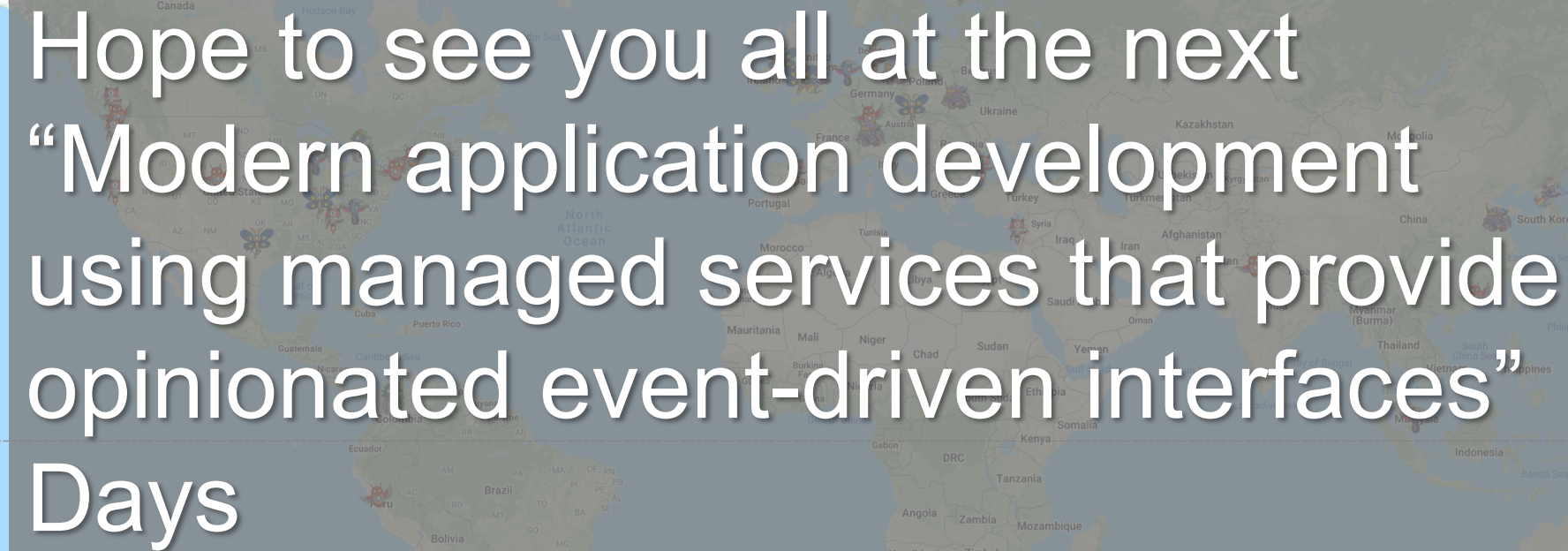




Serverless is Dead!

Long live Serverless!

Photo by [Glenn Carstens-Peters](#) on [Unsplash](#)

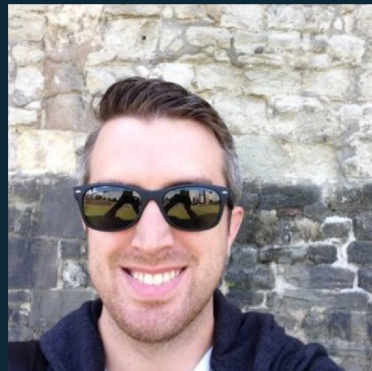


Hope to see you all at the next
“Modern application development
using managed services that provide
opinionated event-driven interfaces”
Days

About me:

Chris Munns - munns@amazon.com, [@chrismunns](https://twitter.com/chrismunns)

- Principal Developer Advocate - Serverless
- New Yorker
- Previously:
 - AWS Business Development Manager – DevOps, July '15 - Feb '17
 - AWS Solutions Architect Nov, 2011- Dec 2014
 - Formerly on operations teams @Etsy and @Meetup
 - Little time at a hedge fund, Xerox and a few other startups
- Rochester Institute of Technology: Applied Networking and Systems Administration '05
- Internet infrastructure geek



DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO
DANKE MERCI THANK YOU GRACIAS ARIGATO

Chris Munns

munns@amazon.com

@chrismunns



AC...

QUESTION
EVERYTHING

"
TEASE
TLW