CLOUD DAY 2024

improove

Milano, Nov 20

Serverless, lo Stato dell'Unione

Luca Bianchi

CTO Talks

Kudos

CLOUD DAY 2024

Milano, Nov 20

improove

hello

I am the one on the right!

Luca Bianchi, PhD

CTO @Neosperience Health, proud dad, and AWS Serverless Hero, passionate about software architectures, serverless, and machine learning.

Serverless Italy, [Gen]AI Italy, and MMUG Meetup co-founder.

ServerlessDays Milano and AWS Community Day co-organizer.

github.com/aletheia

https://it.linkedin.com/in/lucabianchipavia

@bianchiluca

https://speakerdeck.com/aletheia

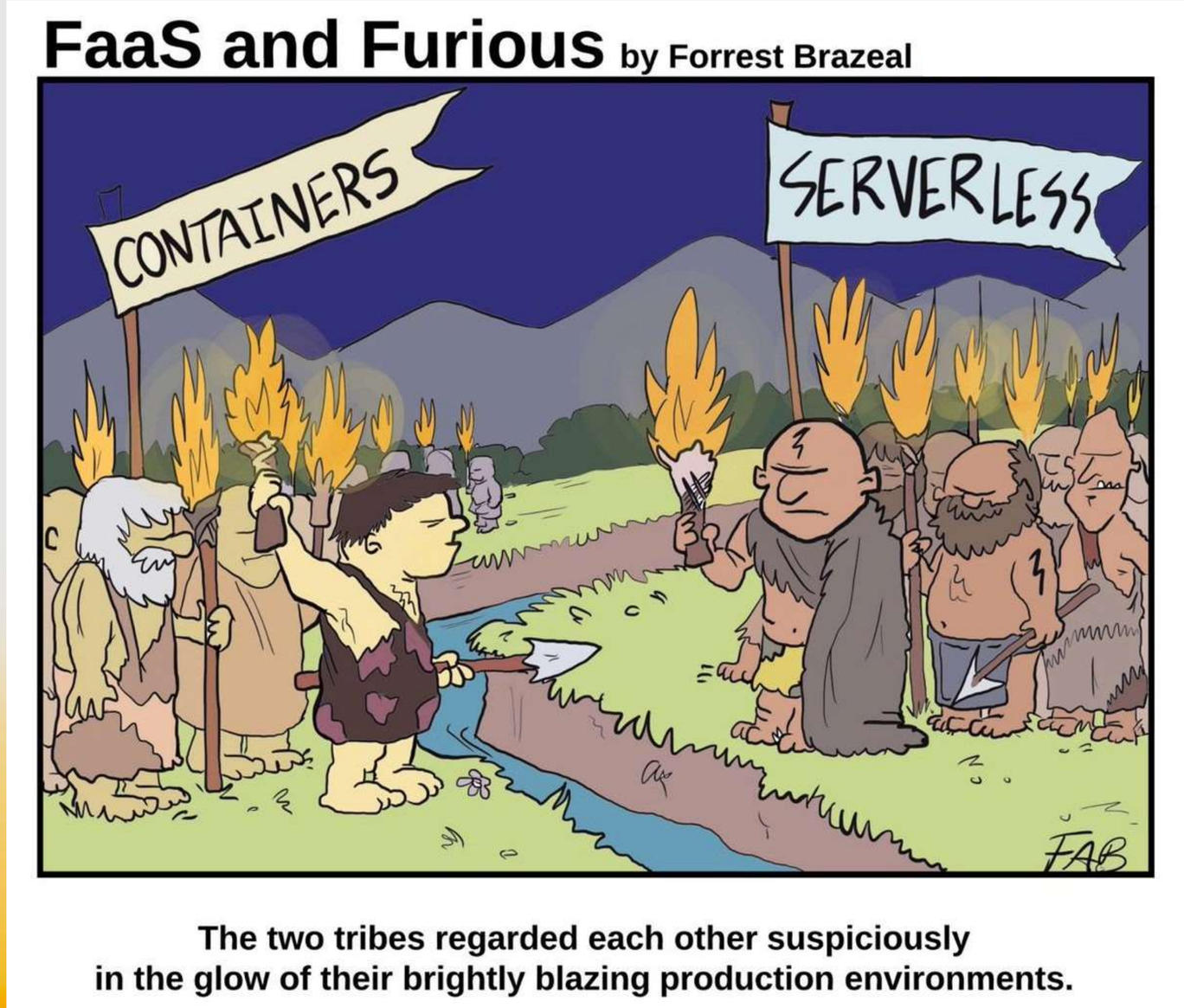
bianchiluca.com

how did it start?


Today, We're Adding These Primitives to AWS with
a New Compute Service…

What is not serverless?

- serverless aren't containers
  - or at least we used to think so
- they have different capacity management tools
- they have different provisioning tools
- they use different packaging
  - such as a zip file
  - and docker images (2020) — ouch!!



What is serverless?

"Serverless architecture replaces long-running virtual machines with ephemeral compute power that comes into existence on request and disappears immediately after use.

Use of this architecture can mitigate some security concerns such as security patching and SSH access control, and can make much more efficient use of compute resources. These systems cost very little to operate and can have inbuilt scaling features."

— ThoughtWorks, 2016

how will it end?

Gregor Hohpe

Gregor Hohpe • 1st
Strategist, engineer, author, speaker. Likes cloud and distribute...
View my blog
3mo •

All good things (and the bad ones--see screen capture below) come to an end.
As I resigned from AWS today, many things are on my mind... no, ser[illegible] isn't dead (neither are zombies--had to squeeze in one bad jok[illegible] and [illegible] also [illegible] en[illegible] chan[illegible]el, s[illegible] always had a weakness for U[illegible] anyhow.

I will stay true to my interests in #CloudAutomation, #ArchitectureAsCode, yes, #Serverless (can't miss all the exciting debates), #PlatformEngineering, and #CarMetaphors. I have a lot to write and update on Enterprise Integration Patterns (.com) and look forward to hanging out with many folks at upcoming events (API Days, Devoxx, NDC, STACK, QCon).

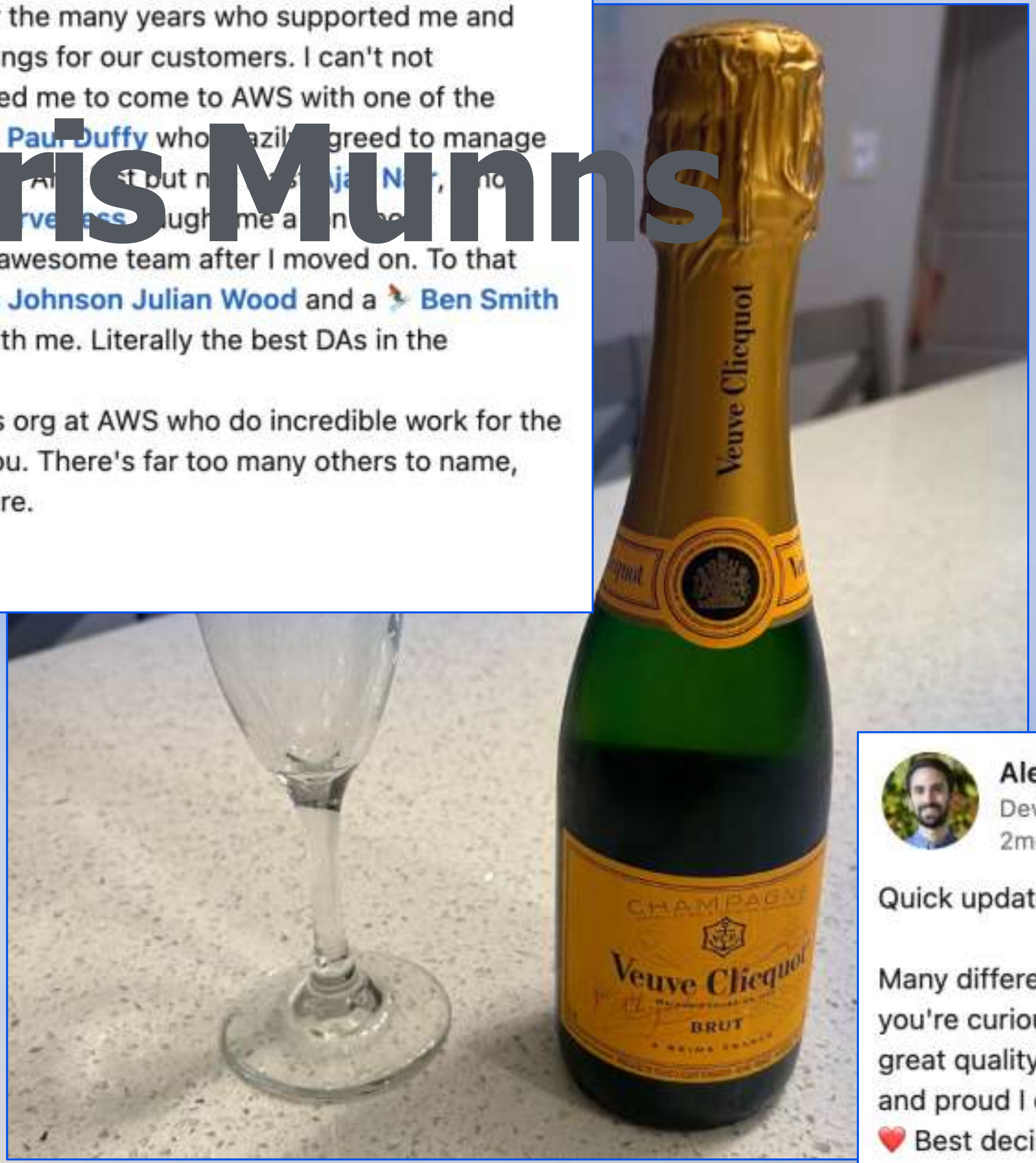
Chris Munns

Chris Munns • 1st
Product Manager - Observability @ Elastic
2mo •

Today was my last day at Amazon/AWS. After 12+ years in total I've decided to take on a new adventure and will be starting my new gig in a few weeks. No badge on laptop photos, no Day 1 quotes. It's been a lot of a lot and im grateful to friends along the way.

I want to specifically thank peers over the many years who supported me and worked with me to deliver on great things for our customers. I can't not thank Paul Horvath who first convinced me to come to AWS with one of the most convincing lines I['ve ever] heard. Paul Duffy who [amaz]ingly agreed to manage me not once, but twic[e] over a [illegible] supported the crazy id[eas] of [illegible] for [Se]rverless [illegible] leadership, and helped champion my awesome team after I moved on. To that point a hat tip to James Beswick Eric Johnson Julian Wood and a 🏃 Ben Smith (among others)who rode that ride with me. Literally the best DAs in the industry. Incredible chaps at that!
Lastly my recent peers in the Startups org at AWS who do incredible work for the hottest companies out there, thank you. There's far too many others to name, and I'm sorry I can't fit everyone in here.

Cheers!



Heitor Lessa

Heitor Lessa • 1st
Principal Engineer
3mo • Edited •

After 11 years, this was my last week at Amazon.

I cannot overstate how privileged I was to be part of the ...more

James Beswick

James Beswick • 2nd
+ Follow
Building Developer Advocacy, ex-AWS
5mo •

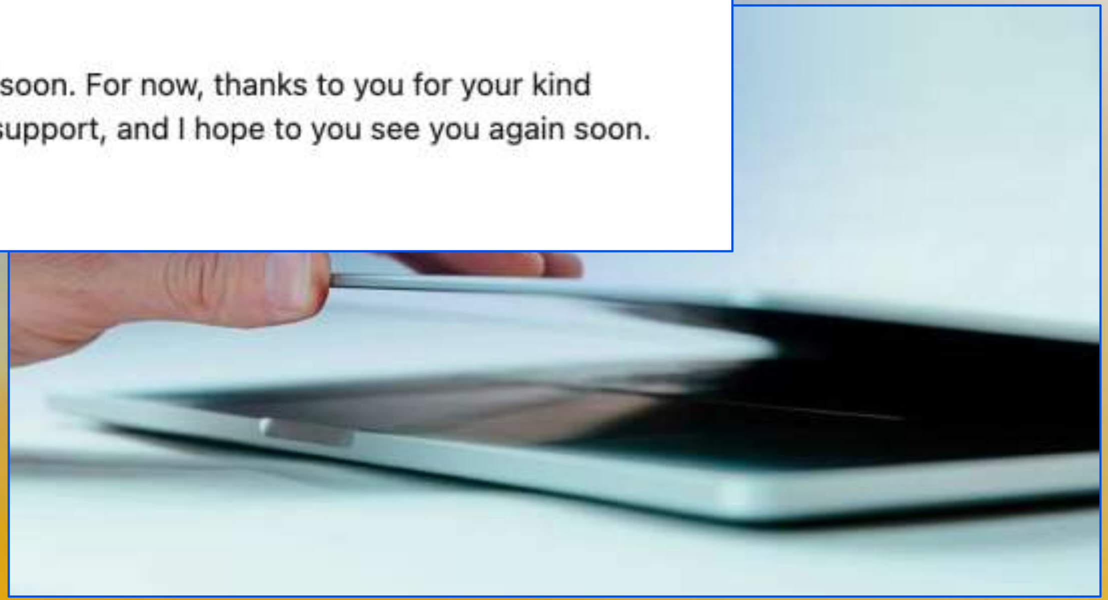
Hey everyone! I'm leaving Amazon Web Services.

It's been awesome to lead the Serverless Developer Advocate team that Chris Munns started. This has been the best team in my career, and truly had some of the smartest, passionate, and kindest people I've worked with. Our small group was able to reach a huge audience, and our creations from Serverless Land to Serverlesspresso and ServerlessVideo were born directly from the questions we heard every day.

After 5 years, it's amazing to see how the strength of the serverless community has grown. It's a g[reat] [illegible] of smart technologists who are generous with their time, buildin[g] [illegible] an enormous a[mount] [illegible] the community, and I hope we gave back to you in equal amount.

It's time to start my next day 1 soon. For now, thanks to you for your kind attention over the years, your support, and I hope to you see you again soon.

#serverless #aws



Alex Casalboni

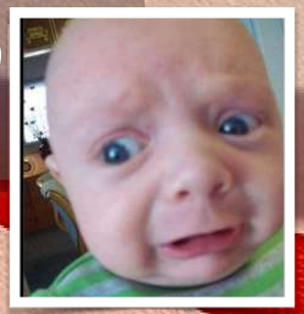
Alex Casalboni • 1st
Developer Advocate @ Edgee | Passionate about sharing & helpi...
2mo •

Quick update to share an (old) news: I left AWS 3 months ago 👋

Many different reasons and personal thoughts about it (do reach out if you're curious). But the important bit is that I've been spending some great quality time with family, especially our 1yo boy. And I'm so happy and proud I could be there every day [when] he started walking and speaking
❤️ Best decision of my life so far!

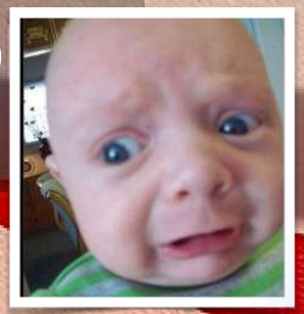
when I saw this..

OMG! is Lambda leaving AWS??

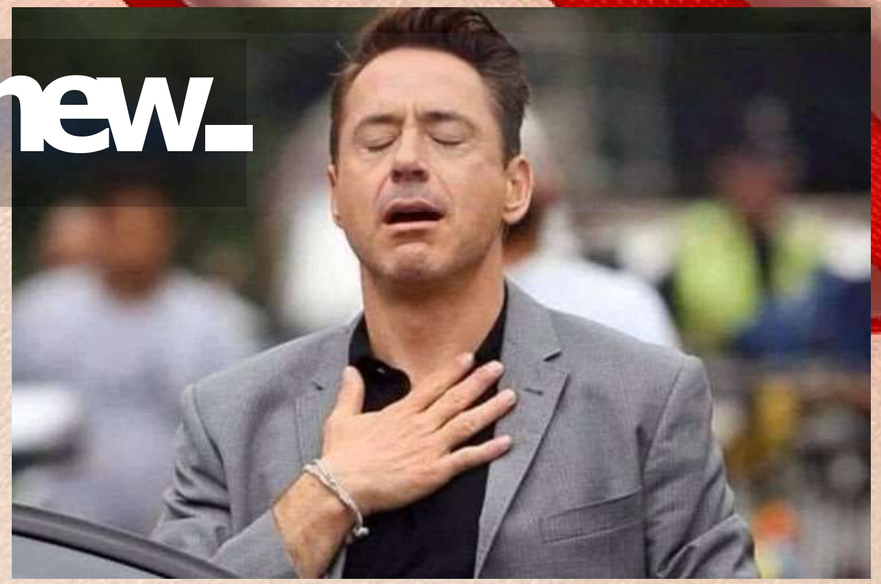


when I saw this..

OMG! is Lambda leaving AWS??



phew..



Werner Vogels • 2nd
VP & CTO at Amazon.com
5d •

✓ Following ···

As we mark #AWS Lambda's 10th anniversary, I'm sharing something we rarely do: the internal PR/FAQ doc that launched this transformative service. It's a fascinating glimpse into our vision for serverless computing and the challenges our customers faced in the early 2010s — and underscores how crisp writing and a deep understanding of customer needs can shape groundbreaking products.

Read the blog post and PR/FAQ here:

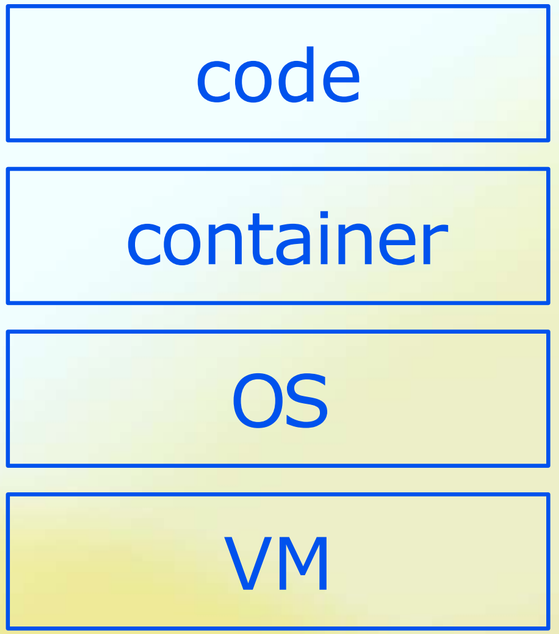
AWS Lambda turns 10: A rare look at the doc that started it
allthingsdistributed.com

still trying to figure out what serverless is..

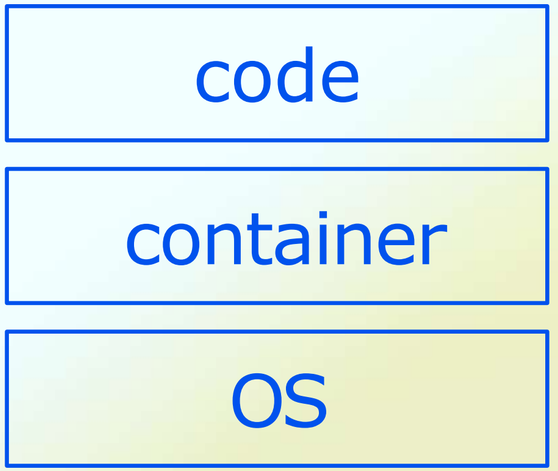
serverless is Function-as-a-Service

Serverless is **no servers**

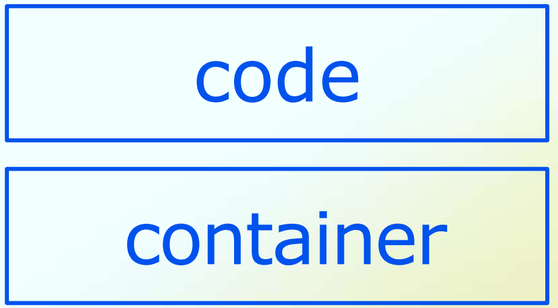
Serverless is no VM



Serverless is no OS to manage



Serverless is **no container**



Serverless is Servicefull

Patrick Debois - 2016

Serverless is Servicefull

Patrick Debois - 2016

serverless language support

	aws Lambda	Azure Azure Functions	Google Cloud Cloud Functions
Runtimes			
Custom (Linux)	Yes	Yes	No
Custom (Windows)	No	Yes	No
Python	Yes	Yes	Yes
Node.js	Yes	Yes	Yes
PHP	No	No	Yes
Ruby	Yes	No	Yes
Java	Yes	Yes	Yes
.NET	Yes	Yes	Yes
Go	Yes	Yes	Yes
Rust	Yes	Yes	No
C/C++	Yes	Yes	No

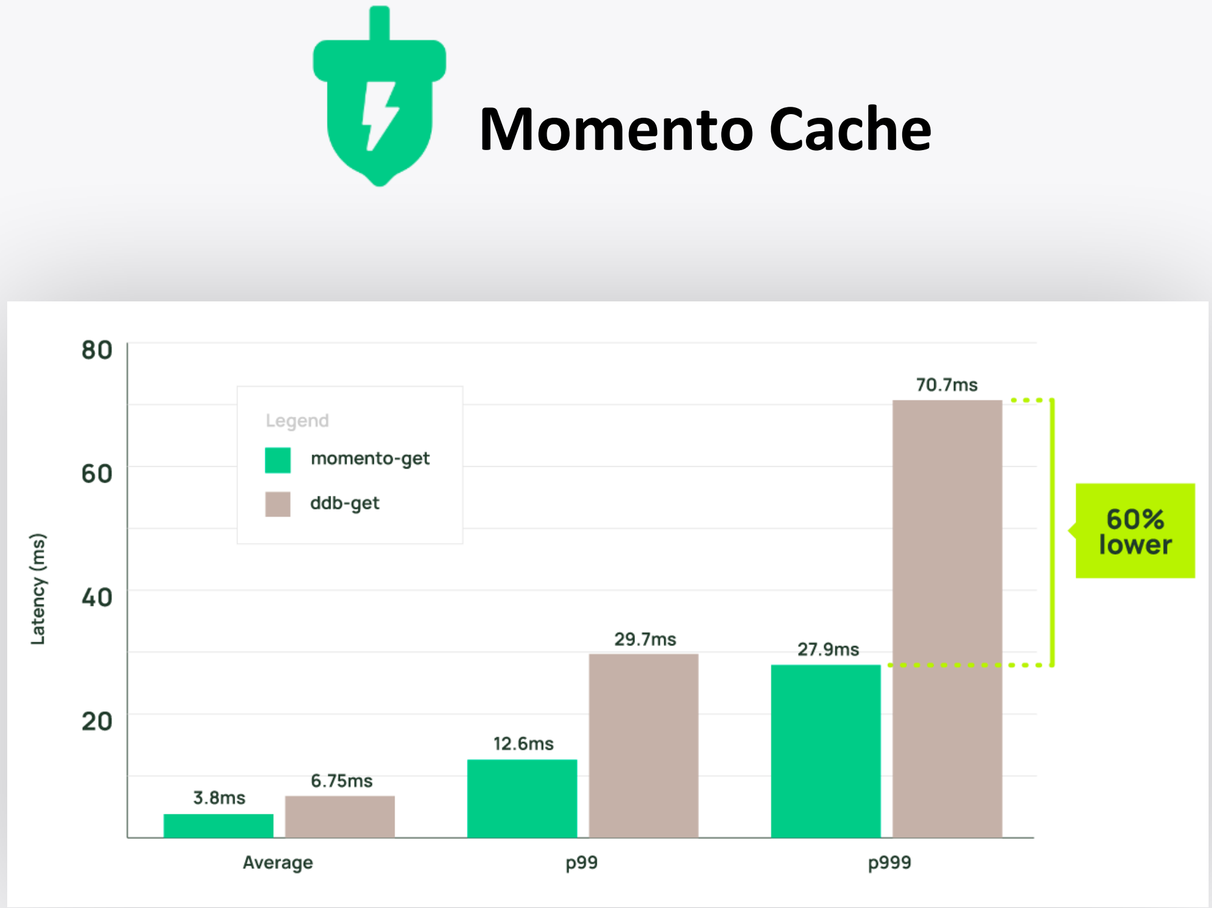
serverless is beyond FaaS

Modern Serverless Capabilities

	AWS	Azure	Google
Arm64	Yes		
Wildcard SSL certificate free	Yes	No	Yes
Serverless KV store	DynamoDB	CosmosDB	Datastore
Serverless SQL	Aurora Serverless	Azure SQL	BigQuery
IaC	SAM, CloudFormation, CDK	ARM, Bicep	GDM
IaC drift detection	Yes		No
Atomic deployment	Yes	No	No
GPU	ECS, EKS	GPU Functions	CloudRun, GKE

Serverless Cache

- DynamoDB is sometimes used as a serverless cache (providing single-digit millisecond response).
- ElastiCache (with Redis) but it is not truly serverless.
- Momento Cache is a fast and easy-to-use cache with SDK.



Serverless Databases

Several solutions with modern features:

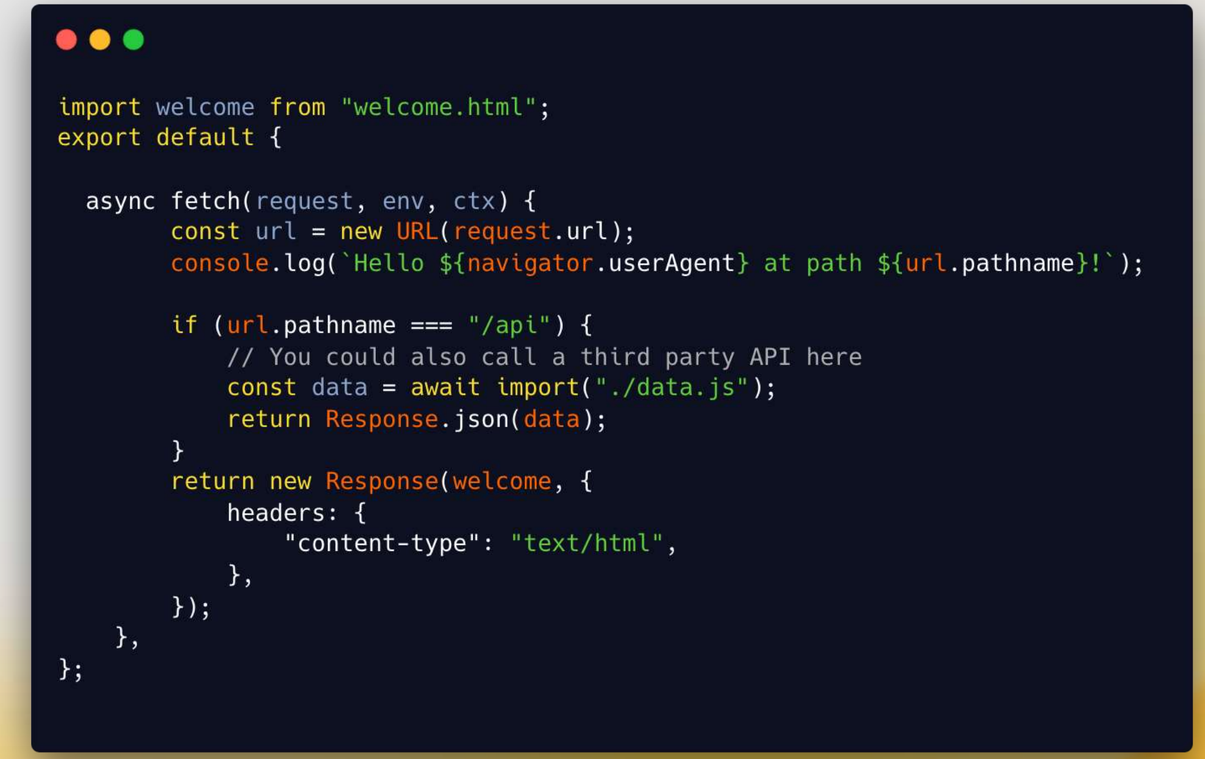
- Real-time Access (low latency)
- Infinite Scalability, High Security, Availability
- Schemaless / Relational?
- Function Deployment

Vendor offering is wide and expanding:

- AWS Dynamo DB
- Amazon Aurora Serverless
- Azure CosmosDB
- BigQuery / Redshift

Edge deployments

- Deploy FaaS at edge
- Transparent multi-region support
- Supports JS, Rust, C, C++
- Key/Value Edge storage
- Lambda @Edge on steroids



```
import welcome from "welcome.html";
export default {

  async fetch(request, env, ctx) {
        const url = new URL(request.url);
        console.log(`Hello ${navigator.userAgent} at path ${url.pathname}!`);

        if (url.pathname === "/api") {
            // You could also call a third party API here
            const data = await import("./data.js");
            return Response.json(data);
        }
        return new Response(welcome, {
            headers: {
                "content-type": "text/html",
            },
        });
    },
};
```

Serverless 101 FAQs

Ten years later, and still wondering..

How micro is a microservice?
It depends. Decompose your system into domain specific computing units using Domain Driven Development (DDD)

Do we want to reinvent the wheel?
AWS provides a variety of managed services that can ease out software development, reducing time to market of orders of magnitude.

How to deal with the outside world?
Rely on web standards and events. Decouple everything.

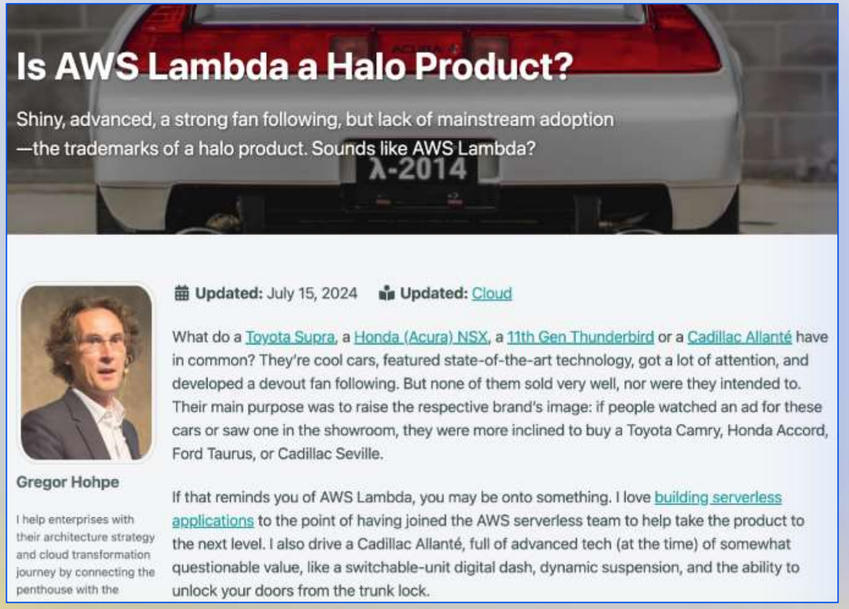
How about vendor lock-in?
Serverless does not lock you in. Data does. But it's the same with languages, tools or frameworks.

serverless is a community

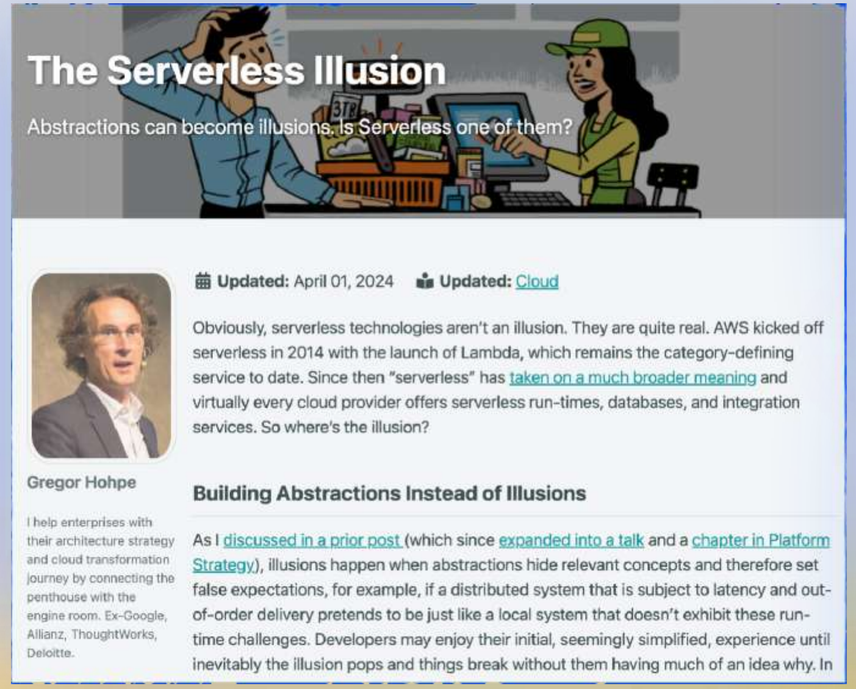
Serverless communities

	aws Lambda	Microsoft Azure Azure Functions	Google Cloud Cloud Functions
Community			
Reddit community members	278,455	141,924	46,415
Stack Overflow members	256,700	216,100	54,300
Videos on YouTube channel	16,308	1,475	4,750
Twitter/X followers	2.2 M	1 M	533 K
GitHub stars for JS SDK	7.5 K	1.9 K	2.8 K
GitHub stars for .NET SDK	2 K	5 K	908
GitHub stars for Python SDK	8.7 K	2.7 K	4.6 K
GitHub stars for Go SDK	8.5 K	1.5 K	3.6 K

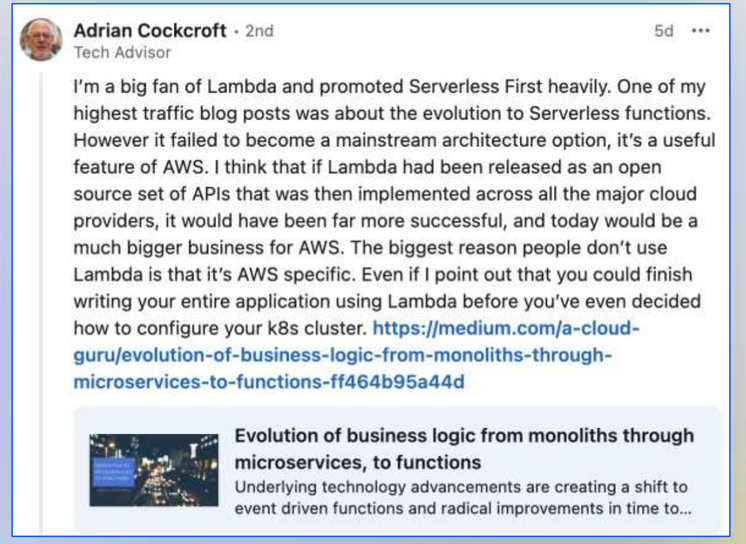
is serverless dead?

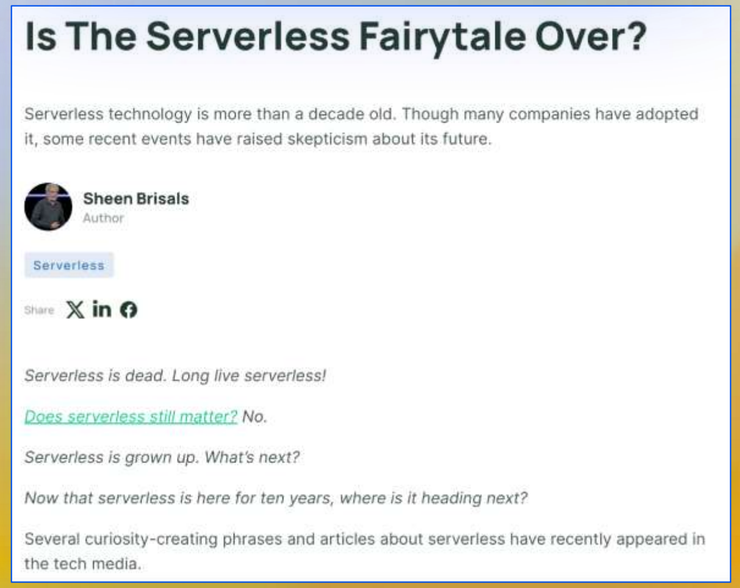


https://architectelevator.com/cloud/lambda-is-a-halo-product/



https://architectelevator.com/cloud/serverless-illusion/



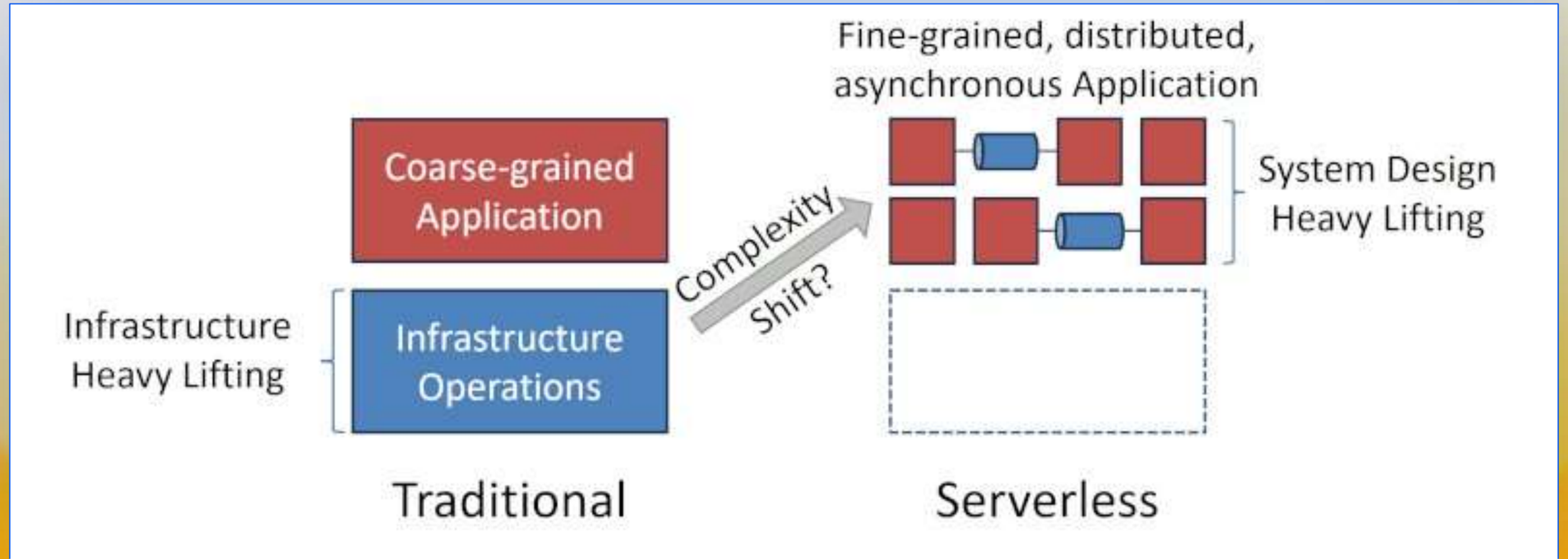


is serverless an illusion?

No. It's an abstraction. Use it wisely

- **The Run-Time illusion**
  Serverless brings a whole new set of things to cope with: cold starts, throttling, etc.

- **The Simplicity Illusion**
  Building serverless solutions reduces operational effort to manage resources. But coding such applications is complex as you're facing the realities of fine-grained, asynchronous systems



A fine-grained, asynchronous application is likely more resilient and cost-effective than a monolith running on VMs.

But it may be harder to build and operate.

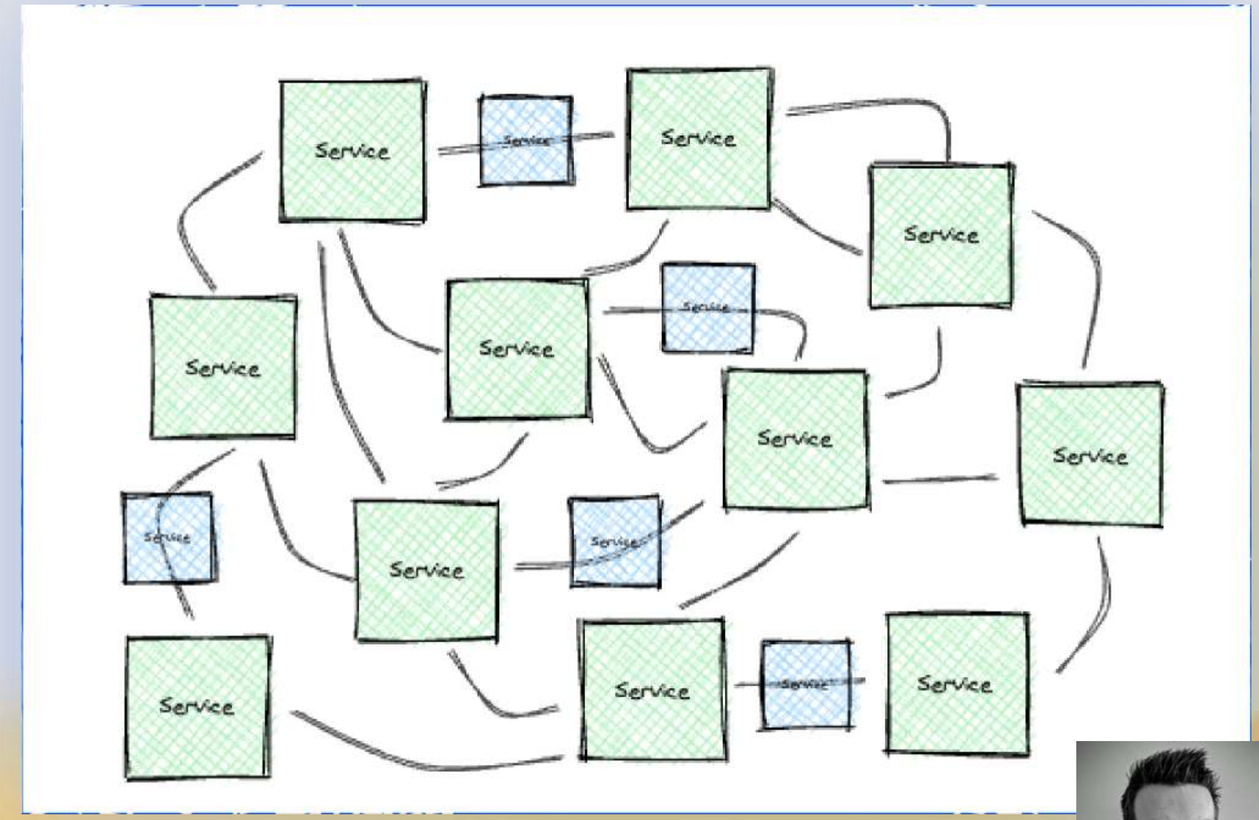
serverless is a community
questioning serverless

serverless isn't a runtime
it's an architecture

AVOIDING THE BIG BALL OF MUD IN EDA

Big Ball of Mud

- Architectures grow to the point they have unclear boundaries.
- Models are unclear within the architecture, hard to understand and change.
- Architecture can turn into a spaghetti code mess.
- With event-driven architecture it can become quite easy to fall into this type of architecture.

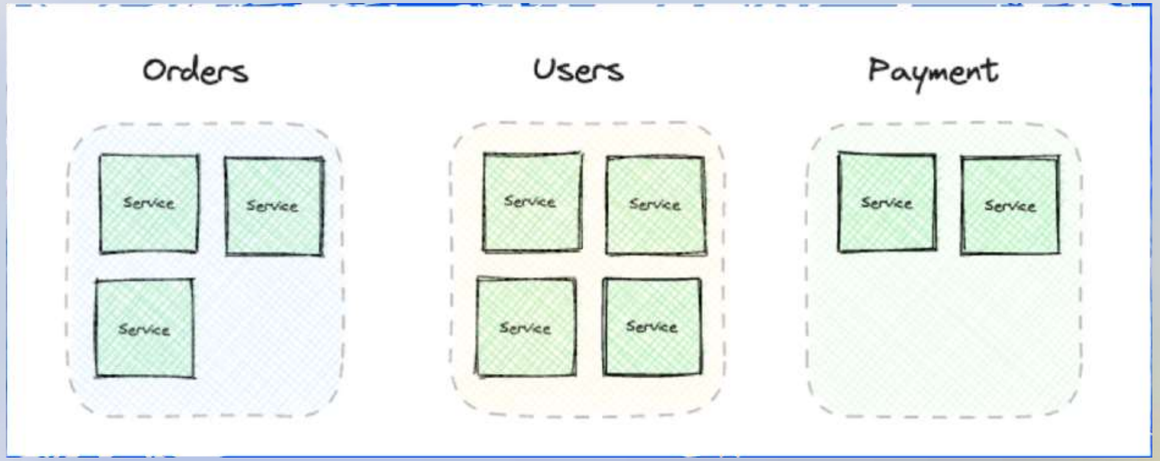


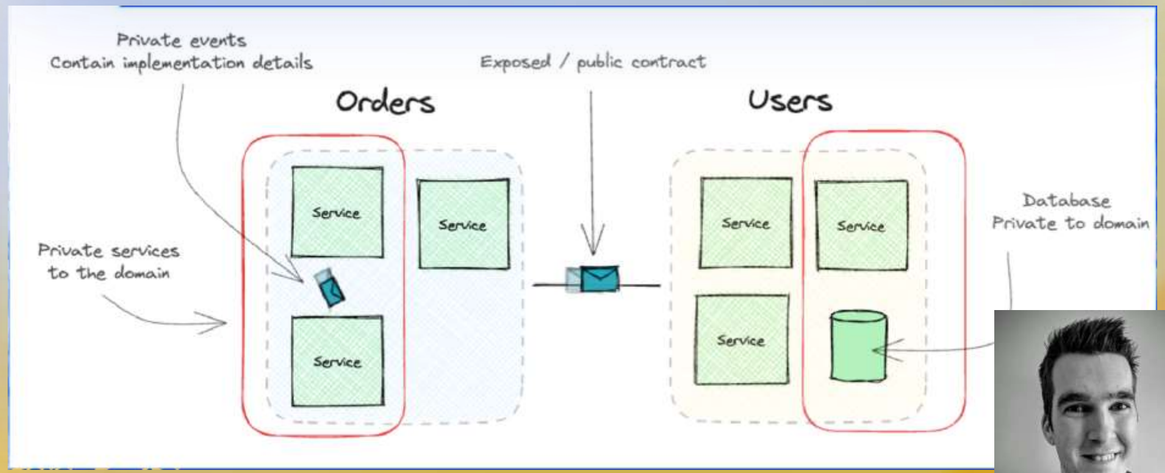
by David Boyne

AVOIDING THE BIG BALL OF MUD IN EDA

follow architecture principles

- Create explicit domains and boundaries
  - using event storming or event modeling
- Understand private/public information.
- Define a language and use context mappings





by David Boyne

maybe it’s time for a new buzzword..

Refactoring to serverless-native

Shifting code from application to automation is what we call "Refactoring to Serverless". Refactoring is a term popularized by Martin Fowler in the late 90s to describe the restructuring of source code to alter its structure without changing its external behavior. Code refactoring can be as simple as extracting code into a separate method or more sophisticated like replacing conditional expressions with polymorphism.

Developers refactor their code to improve its readability and maintainability. A common approach in Test-Driven Development (TDD) is the so-called *red-green-refactor* cycle: write a test, which will be red because the functionality isn't implemented, then write the code to make the test green, and finally refactor to counteract the growing entropy in the codebase.

Serverless refactoring takes inspiration from this concept but augments it to the context of serverless automation:

> *Serverless refactoring: A controlled technique for improving the design of serverless applications by replacing application code with equivalent automation code.*

serverless-native refactoring

Refactoring to Serverless

3 MIN

Serverless is more than just a run-time for your code. It's a suite of cloud provider managed services to help fine-grained, event-driven applications best utilize the cloud. Automation is an integral part of building serverless applications and often the same functionality can be achieved from application code or through a platform feature. Take sending a message from a Lambda function to SQS as an example. You could do this by writing code, utilizing a client library like `boto3` or you could configure a Lambda Destination. The latter approach is preferred because it utilizes the cloud platform and separates application topology (which component talks to which others) from the application logic. Interestingly, with AWS CDK you can configure the Lambda destination using the same programming language that you used to write your application code.

> *"You can improve the design of your serverless application by replacing application code with automation code, while using the same programming language."*

That's what we call *Refactoring to Serverless*, based on Martin Fowler's definition of Refactoring as a popular coding technique to improve your application design:

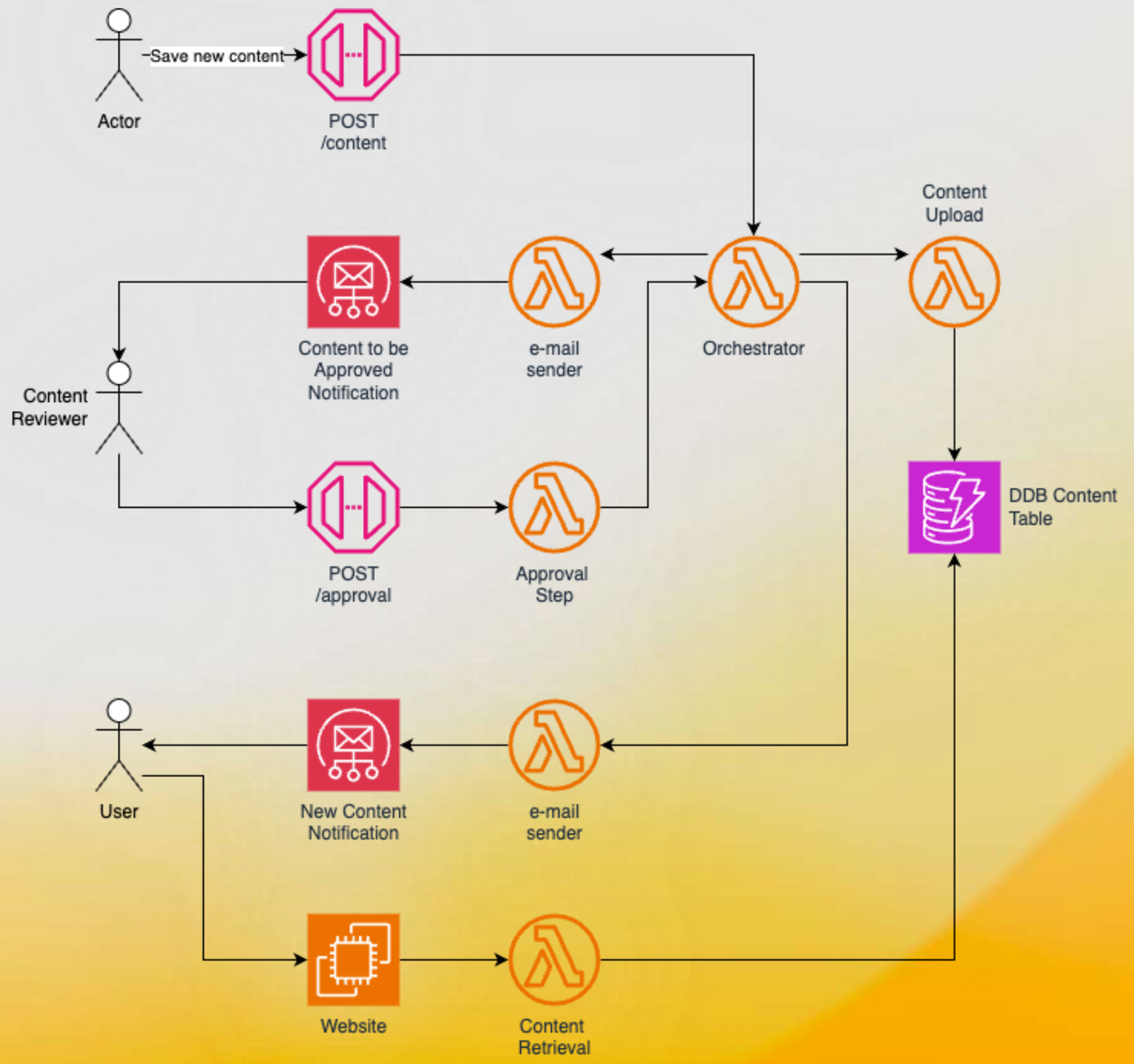
> *"A disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior"*

In our case, the refactoring moves code from the application to CDK automation. Other refactorings might improve the overall application design. Let's look at the example in more detail.

Serverless Newsletter

2015

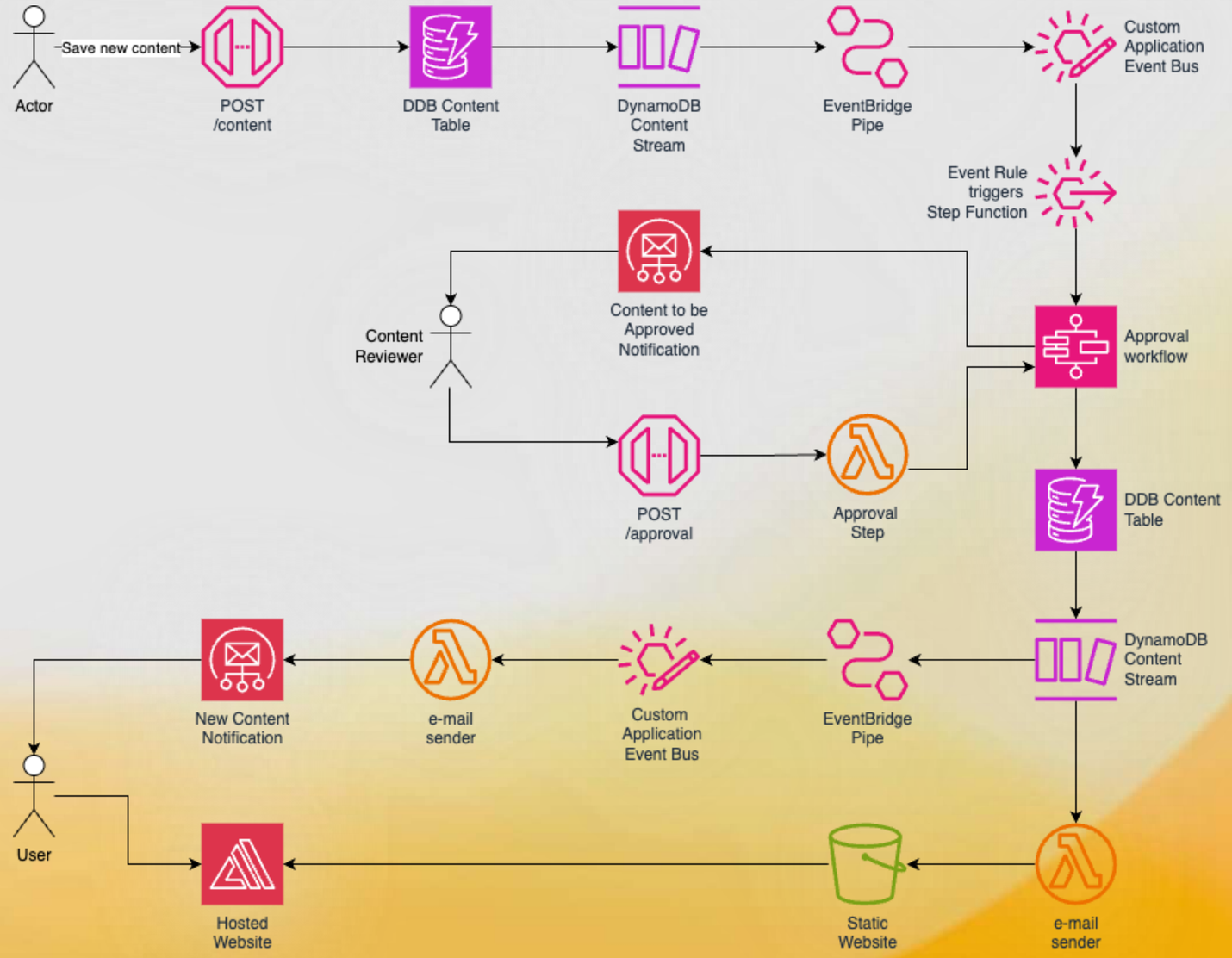
- Single point of failure.
- Everything is synchronous.
- Service coupling.
- Difficult to test.



Serverless Newsletter

2024 — refactored

- No single point of failure.
- Logic is testable
- Subsystems can be reused.
- Easy to test.
- Many moving parts.



serverless is convergence

Serverless means modern applications

modern applications so far..

- FaaS
- serverless managed services
- Infrastructure-as-Code
- event-driven-architectures (EDA)

Infrastructure-from-Code

abstract away cloud resources

- Developers just need to write business code.
- Infrastructure is inferred from API calls and information flow configuration.
- Framework maintains feature parity with cloud vendors.

```
import { api, data, events } from '@some-ifc-sdk'

api.post("/users", async (req, res) => {
  const { email, name } = req.body;
  const newUser = await data.set(`user:${email}`, { email, name });
  res.send({ user: newUser });
});

data.on("created:user:*", ({ item }) => {
    console.log("New user created!");
    events.publish("user.created", { after: "1 day" } item)
});

events.on("user.created", (event) => {
    console.log('user.created event received!');
    // Send a follow up email, call an API, etc.
})
```

this code translates into

Ampt

- Infrastructure is built around your code from SDK calls.
- Ampt SDK exposes abstractions for storage, tasks, APIs, parameters, and web sockets.
- Supports frameworks such as Express, NextJS, Nuxt, NestJS, React, Angular, Astro, SvelteKit, etc.
- Offer support to frontend deployment with no configuration.

```
import { api } from "@ampt/api";
import {data } from "@ampt/data";

// define a public api and create a new router
const myApi = api("ampt-demo").router("/demo");

myApi.post("/hello", async (event) => {
  const body = await event.request.body(); // { name: "Luca" }
  const { name } = body;
  await data.set("name", name);
  return event.status(200).body({ message: `Hello ${name}` });
});

myApi.get("/hello", async (event) => {
  const name = await data.get("name");
  return event.status(200).body({ message: `Hello ${name}. I've
retrieved your name from storage` });
});
```

Wing

a programming language for the cloud

- Combine infrastructure and runtime in one language
- cross-cloud support
- iterate and test locally with the simulator
- end-to-end testing framework
- $20M in funding
- landing zone of some of the top serverless advocates (Elad Ben-Israel, David Boyne)
- Featured in Gartner Hype Cycle for Platform Engineering

```
bring cloud;

// define an API
let api = new cloud.Api();
// define storage
let store = new cloud.Bucket();

// create routes for the API
api.get("/employees", inflight (req) => {
  let result = MutJson [];
  let var i = 0;
  for employee in store.list() {
    result.setAt(i, employee);
    i = i + 1;
  }
  return cloud.ApiResponse {
    status: 200,
    body: Json.stringify(result)
  };
});
```

https://www.winglang.io/

could we make containers more serverless?

Serverless containers

- **Scaleway**
  - container management made easy
  - built-in infrastructure management
- **Google Cloud Run**
- **Azure Container Instances**
- **AWS**
  - Fargate
  - App Runner


How to stay above the clouds?

it’s just day 1

Serverless from zero to hero

https://www.improove.tech/onlinecourses/courses/serverless-on-aws-from-zero-to-hero

Thanks!

CLOUD DAY 2024

Milano, Nov 20

improove