Serverless Java

productivity, security, cost-savings and fun

airhacks.live

airhacks.industries

airhacks.live

"It's not work if you like it"
…so I never worked. #java

airhacks.fm

#185 A Cloud Migration Story: From J2EE to Serverless Java

[episode link] Listen on Apple Podcasts LISTEN ON Spotify Listen on Google Podcasts [RSS]

An airhacks.fm

ZX Spectru
CPC 64, De
in 1993, usi
Lambda, Cl
clouds thei
services, n
quarkus in
the cloud h

#219 Java, CraC and Reducing Cold Start Duration with AWS Lambda SnapStart

[episode link] Listen on Apple Podcasts LISTEN ON Spotify Listen on Google Podcasts [RSS]

An airhacks.fm conv

CRaC API, C1 and
snapshotting the
with MicroProfile
beforeCheckpoin
Zones, SnapStart
SnapStart, The O
Lambda under th
metal,

#240 Serverless Java (17) on AWS

[episode link] Listen on Apple Podcasts LISTEN ON Spotify Listen on Google Podcasts [RSS]

An airhacks.fm conversation with Maximilian Schellhorn (@maschnetwork) about:

playing Halo with Fujitsu Siemens Scaleo, amazing graphics with crytec and crysis, l
marquee tag, semi-professional Call of Duty 4 gaming, learning Delphi and GUI progr
programming in Delphi, building ski school software in Delphi, from Delphi to Java an
with Java, starting at cloudflight.io, from Zalando to AWS, starting at AWS as Solution

airhacks.live

airhacks.TV

with the time machine, “100 episodes ago segment”

…any questions left?

youtube.com/
@bienadam/shorts


500
@bienadam/shorts

airhacks.live

NEW https://discord.gg/airhacks

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Summer 2024:

- Persistence Patterns for Serverless Java on AWS, July, 11th, 2024
- Serverless Generative AI with Java on AWS, July, 25th, 2024

Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by Adam Bien

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io

…I started with DevOps in 1995

then continued with serverless computing in 2001…

Green IT

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

GraalVM™

reduces RAM footprint

https://sites.google.com/view/energy-efficiency-languages/results?authuser=0

motivation / facts

Configure AWS Lambda Info

Architecture

Arm

Number of requests

2

Unit

per second

Duration of each request (in ms)

Duration is calculated from the time your code begins executing until it returns or otherwise terminates.

100

Amount of memory allocated

Enter the amount between 128 MB and 10 GB

Value

2

Unit

GB

Amount of ephemeral storage allocated

Enter the amount between 512 MB and 10,240 MB. The first 512 MB are at no additional charge, you only pay for any additional storage that you configure for the function.

Value

512

Unit

MB

▶ Show calculations

Provisioned Concurrency Info

Total Upfront cost: 0.00 USD
Total Monthly cost: 15.07 USD

Show Details

Save and view summary

Save and add service

Configure AWS Fargate Info

Operating system

Linux

CPU Architecture

ARM

Number of tasks or pods

Enter the number of tasks or pods running for your application

Value

1

Unit

per month

Average duration

Enter the time period for which your tasks or pods are running. Pricing is per second with a 1-minute minimum (Linux) and a 15-minute minimum (Windows). Duration is calculated from the time you start to download your container image (docker pull) until the Task or Pod terminates, rounded up to the nearest second.

Value

30

Unit

days

Amount of vCPU allocated

Enter the amount between 0.25 vCPU and 16 vCPU

1

vCPU selected supports memory values between 2 GB and 8 GB, in 1 GB increments

Amount of memory allocated

8

Unit

GB

Total Upfront cost: 0.00 USD
Total Monthly cost: 50.38 USD

Show Details

Save and view summary

Save and add service

EC2 Instances (625)

Based on your inputs, this is the lowest-cost EC2 instance: **t4g.large**

Chosen instance: **t4g.large** | **Family: t4g** | **2vCPU** | **8 GiB Memory**

Search instance type

Search by instance name or filter by keyword

Instance family Info	vCPUs	Memory (GiB)	Network performance
Any Instance f…	2	8 GiB	Any Network P…

Show only current generation instances.

1 2 3 4 5 6 7 ... 63

	Instance name	vCPUs	Memory	Network Performance	Storage
◉	t4g.large	2	8 GiB	Up to 5 Gigabit	EBS only

Amazon EC2 Compute Savings Plans (Monthly): 33.07

Total Upfront cost: 0.00 USD
Total Monthly cost: 33.07 USD

Show Details ▲ Cancel Save and view summary Save and add service

Configure Amazon RDS for PostgreSQL Info

Selected Instance:

db.m4.large

vCPU: 2

Memory: 8 GiB

Utilization (On-Demand only)

With utilization, you still have to stop the instance to get the cost benefit. Utilization only affects OnDemand pricing for instances and not the storage, backups, etc.

Value: 100

Unit: %Utilized/Month

Deployment Option: Single-AZ

Pricing Model: OnDemand

▼ Show calculations

1 instance(s) x 0.217 USD hourly x (100 / 100 Utilized/Month) x 730 hours in a month = 158.4100 USD

Amazon RDS PostgreSQL instances cost (monthly): 158.41 USD

Amazon RDS PostgreSQL instances cost (upfront): 0.00 USD

Total Upfront cost: 0.00 USD

Total Monthly cost: 188.80 USD

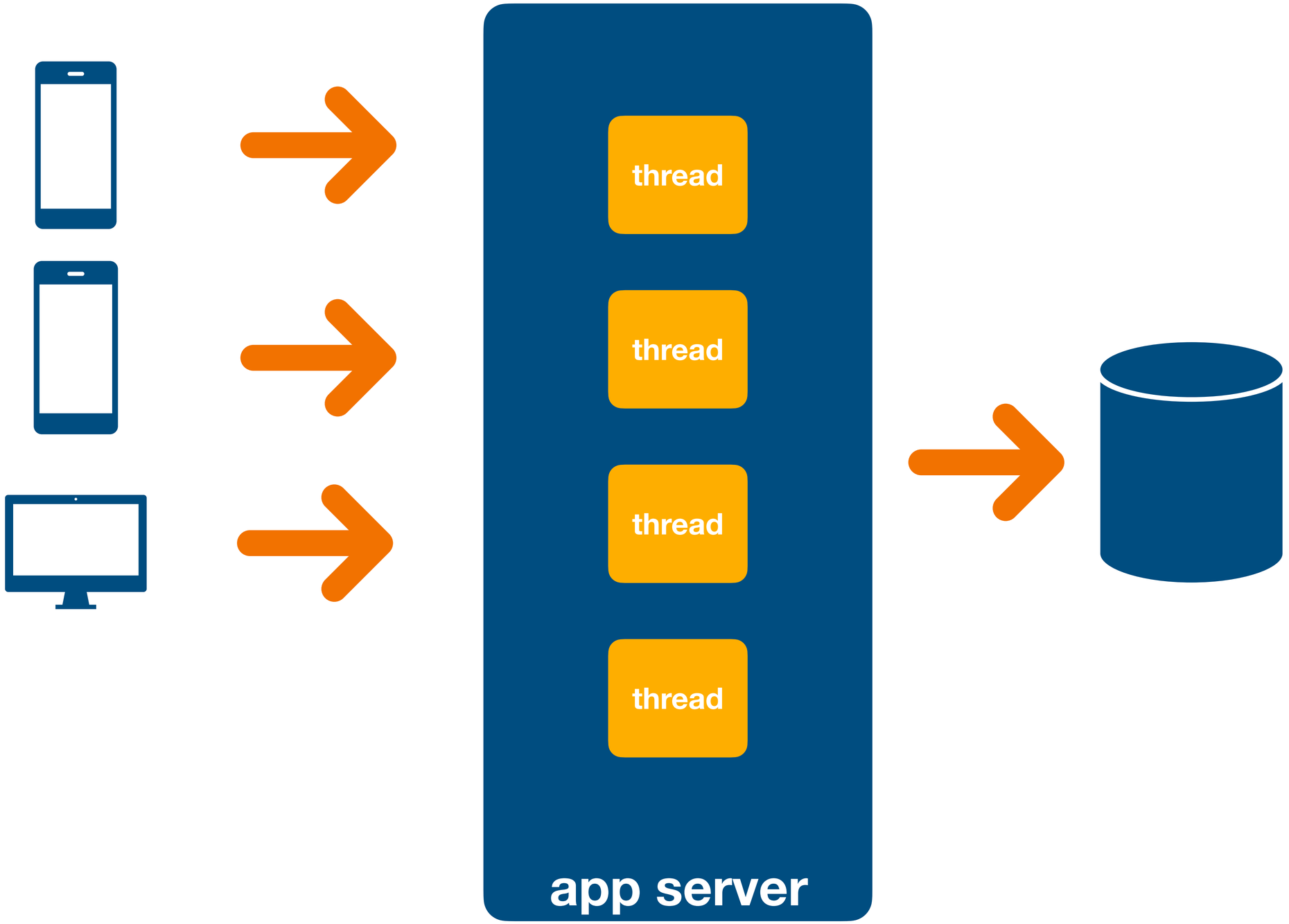
Show Details ▼

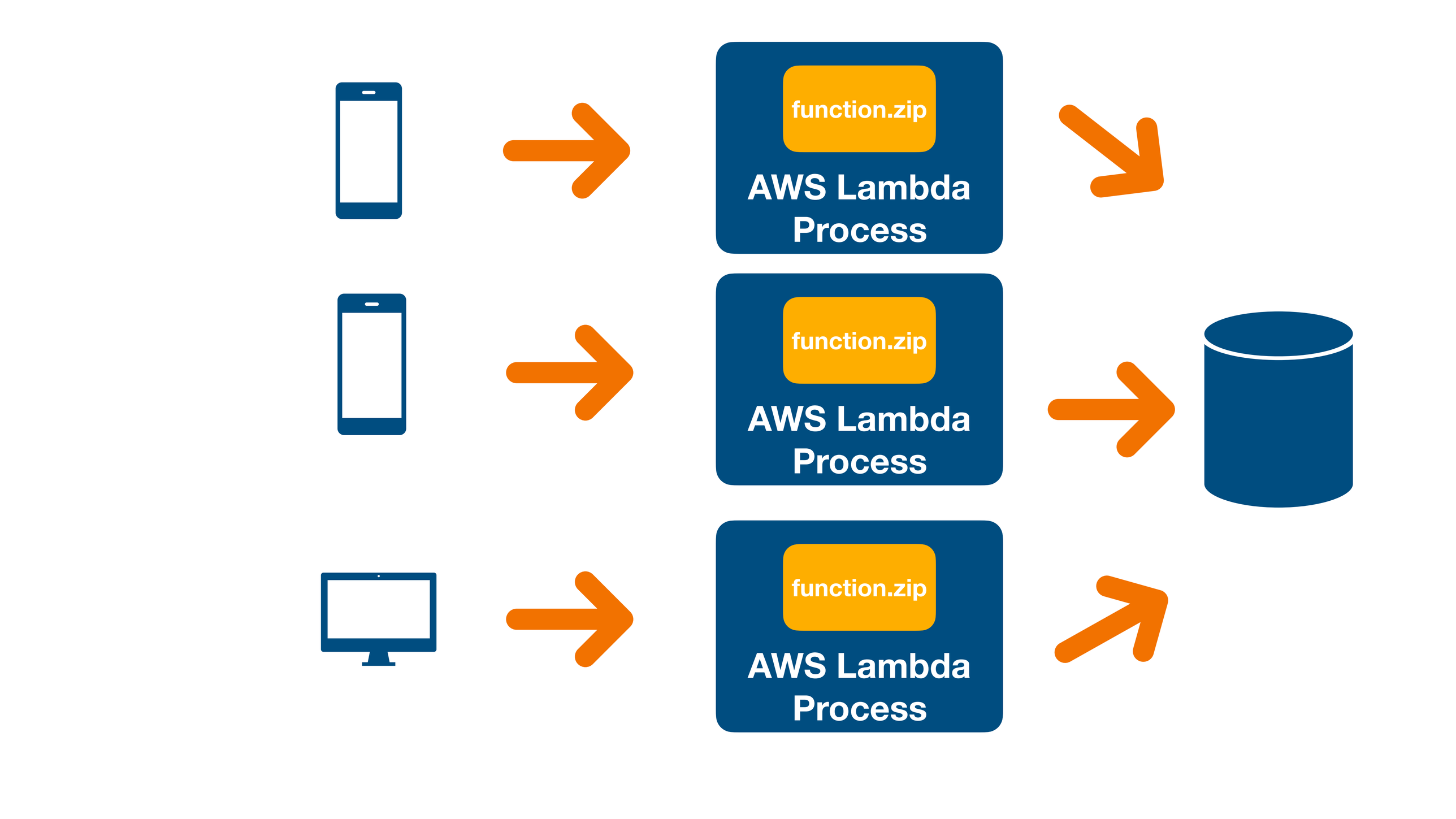
Specifications

Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Gigabit Ethernet
2 USB 3.0 ports; 2 USB 2.0 ports.
Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
2 × micro-HDMI® ports (up to 4kp60 supported)
2-lane MIPI DSI display port
2-lane MIPI CSI camera port
4-pole stereo audio and composite video port
H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
OpenGL ES 3.1, Vulkan 1.0
Micro-SD card slot for loading operating system and data storage
5V DC via USB-C connector (minimum 3A*)
5V DC via GPIO header (minimum 3A*)
Power over Ethernet (PoE) enabled (requires separate PoE HAT)
Operating temperature: 0 – 50 degrees C ambient

The Runtime (R)Evolution


business logic
MicroProfile
AWS Lambda
business logic
MicroProfile
Java SE / bare metal
business logic
MicroProfile
containerized


thread
thread
thread
thread
app server


function.zip
AWS Lambda Process
function.zip
AWS Lambda Process
function.zip
AWS Lambda Process

cloud native / serverless

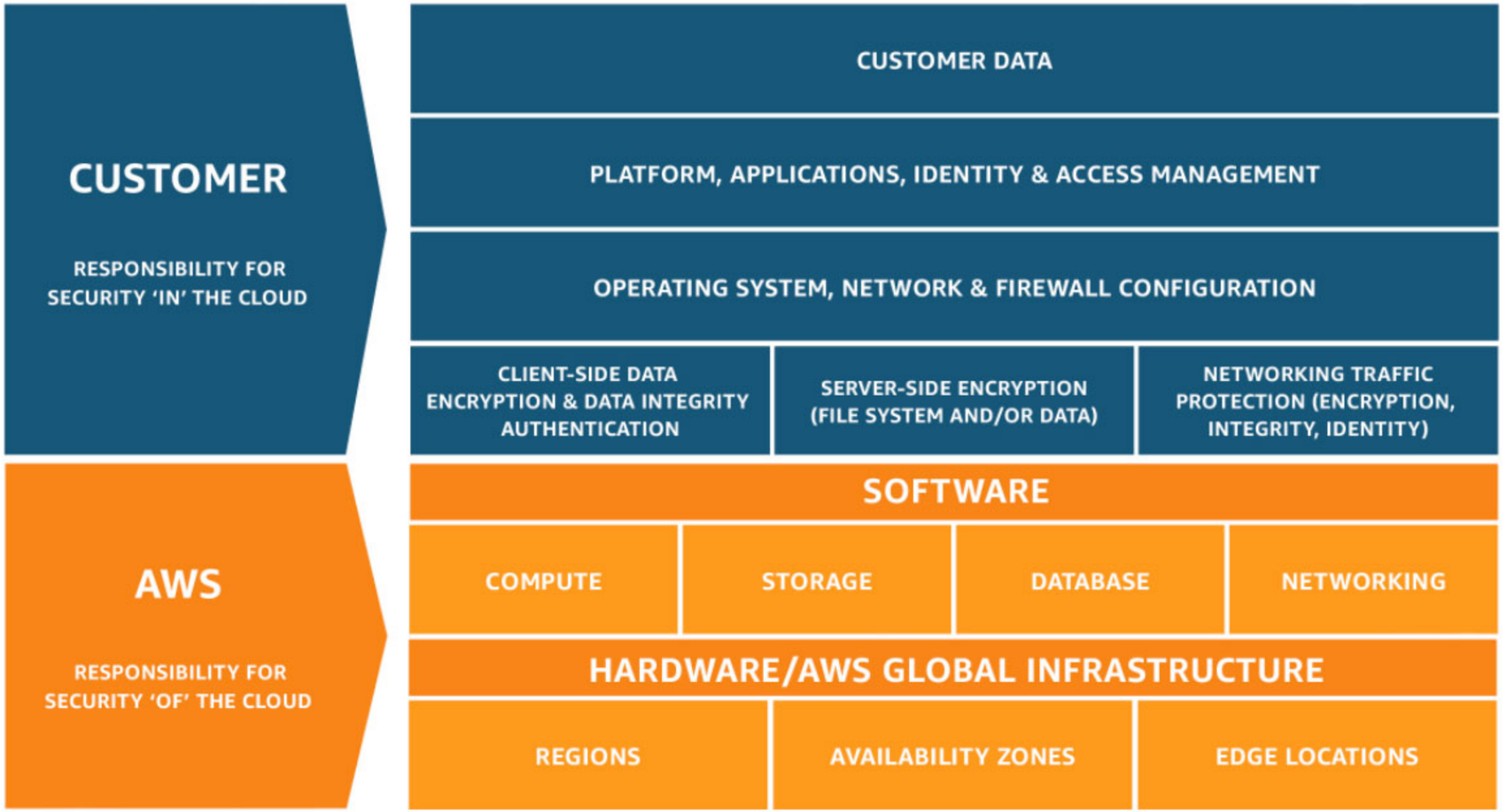
☁️ 👉cost driven architectures

cloud lock-in…

- switching costs

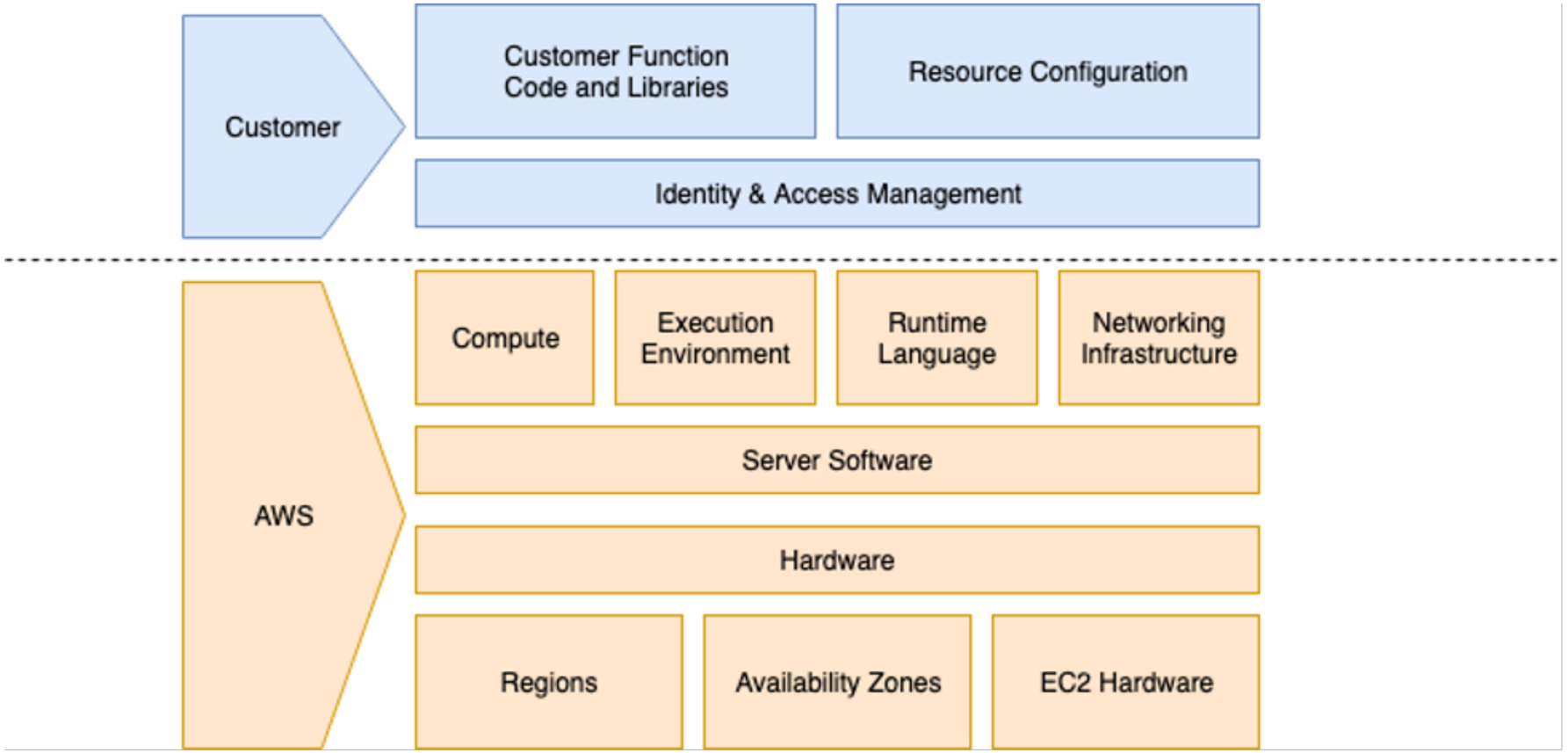
- code portability
- infrastructure portability
- …data portability

boring (like on-premise)…



https://aws.amazon.com/compliance/shared-responsibility-model/

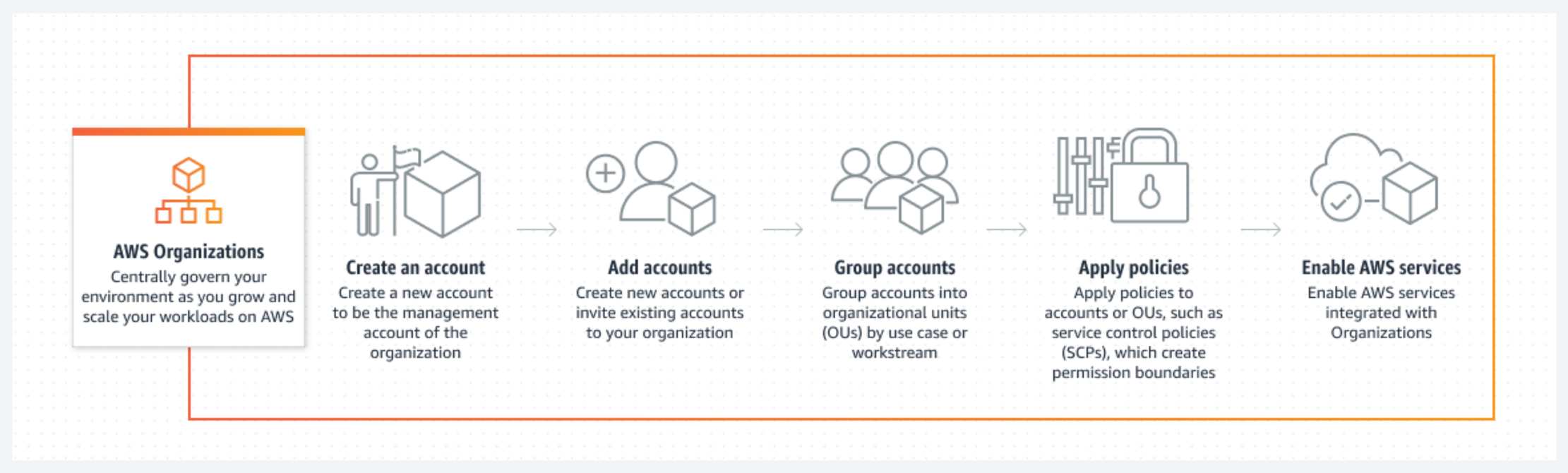
…clouds



https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/the-shared-responsibility-model.html

AWS Organizations

AWS Organizations



https://aws.amazon.com/organizations/

AWS IAM Identity Center

AWS IAM Identity Center

https://aws.amazon.com/iam/identity-center/

Customer Carbon Footprint Tool

DevOps? NoOps!

DEFINITION

NoOps (no operations)

By **Rahul Awati**

What is NoOps (no operations)?

NoOps (no operations) is a concept that an IT environment can become so automated and abstracted from the underlying infrastructure that there's no need for a dedicated team to manage software in-house.

First coined by research and advisory company Forrester, the term describes the goal of NoOps as to "improve the process of deploying applications" so that "application developers will never have to speak with an operations professional again." Forrester's Mike Gualtieri called DevOps "a step backward" even though, in his short 2011 article, he wrote that he admires its goal to improve application release deployment processes.

IaC

Infrastructure as Code

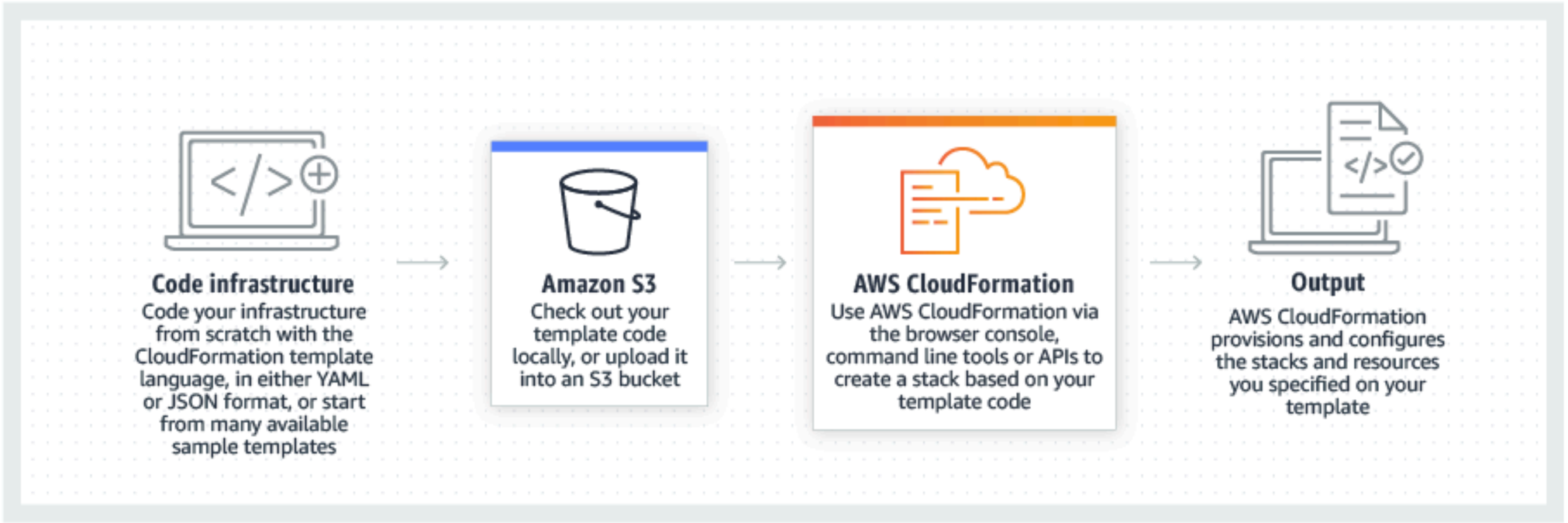
Infrastructure as code (**IaC**) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.[1] The IT infrastructure managed by this process comprises both physical equipment, such as bare-metal servers, as well as virtual machines, and associated configuration resources. The definitions may be in a version control system. It can use either scripts or declarative definitions, rather than manual processes, but the term is more often used to promote declarative approaches.

https://en.wikipedia.org/wiki/Infrastructure_as_code

Infrastructure as Code

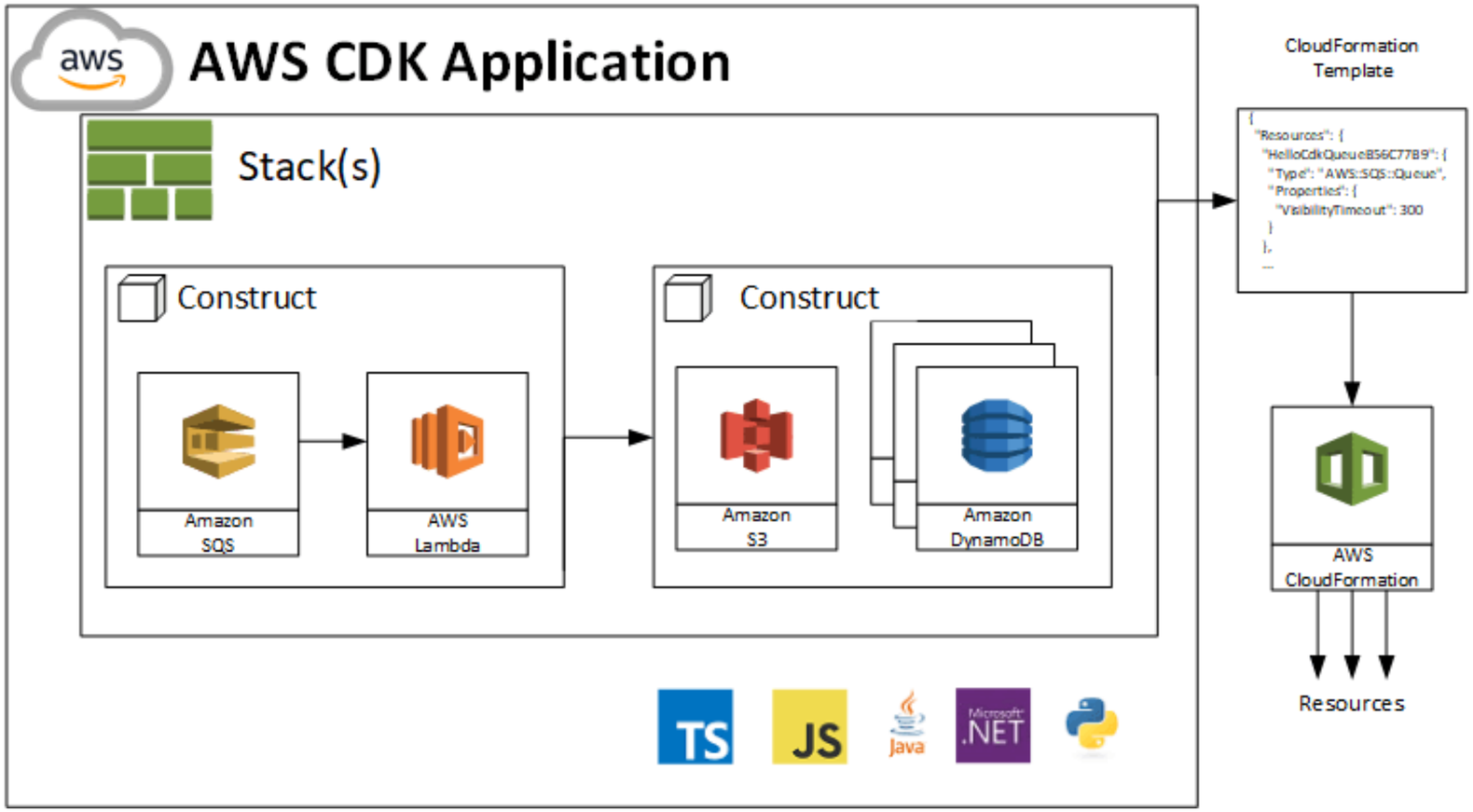
- AWS Command Line Interface (CLI)
- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)

AWS CloudFormation



https://aws.amazon.com/cloudformation/

AWS Cloud Development Kit (CDK)



https://docs.aws.amazon.com/cdk/latest/guide/home.html

Constructs

- L1: Cfn resources
- L2: opinionated resources
- L3: patterns - composite resources
- **custom constructs**

Infrastructure as Code

- “cloud api” integration
- tables, queues and objects are exposed as AWS “Resources”
- no side channels required
- easier IaC
- multi-tenancy out-of-the-box
- integrated authentication and authorization

migrations

recap: cloud lock-in…

- switching costs

- code portability
- infrastructure portability
- …data portability

The 6 R’s

1. Rehosting — Otherwise known as “lift-and-shift.”
2. Replatforming — I sometimes call this “lift-tinker-and-shift.”
3. Repurchasing — Moving to a different product.
4. Refactoring / Re-architecting.
5. Retire — Get rid of.
6. Retain — Usually this means “revisit” or do nothing (for now).

https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/

cloud native services 👉 CDK + SDK

Fat, Monolithic Function

“Nano”, Asynchronous / Event Driven Function

Serverless Archive: SAR (~~EAR~~😊)

App as a Function

if you go to the cloud
…use the cloud!

references

- “Ten Years After: From Java EE 6 to Quarkus and AWS Lambda”: http://youtube.com/@bienadam/
- “Hey Enterprise EJB Developers Now Is The Time To Go Serverless”: http://youtube.com/@bienadam/
- “Past, Present and Future: Serverside Java on Premise and in the Clouds”: http://youtube.com/@bienadam/
- https://github.com/AdamBien/aws-java-functionurl-cdk-plain
- https://github.com/AdamBien/aws-quarkus-lambda-cdk-plain
- https://github.com/AdamBien/aws-cdk-plain
- https://github.com/AdamBien/eras3r
- “Using Amazon Corretto (OpenJDK) for lean, fast, and efficient AWS Lambda Applications” https://aws.amazon.com/blogs/developer/lean-fast-and-oversized-aws-lambdas-on-jvm-amazon-corretto-java/

references

- “Ten Years After: From Java EE 6 to Quarkus and AWS Lambda”: http://youtube.com/@bienadam/
- “Hey Enterprise EJB Developers Now Is The Time To Go Serverless”: http://youtube.com/@bienadam/
- “Past, Present and Future: Serverside Java on Premise and in the Clouds”: http://youtube.com/@bienadam/
- “Saving Costs with Pragmatic Serverless Java Architectures”: http://youtube.com/@bienadam/
- https://github.com/AdamBien/x-ray (~15 years old code)
- https://github.com/AdamBien/lightfish (~15 years old code)

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Summer 2024:

- Persistence Patterns for Serverless Java on AWS, July, 11th, 2024
- Serverless Generative AI with Java on AWS, July, 25th, 2024

Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by Adam Bien

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io

Thank YOU!

airhacks.industries

airhacks.live