UNIVERSIDADE
BEIRA INTERIOR

# Serverless Computing

Computação na Cloud
Mestrado em Engenharia Informática

Mário M. Freire
Departamento de Informática

Ano Letivo 2023/2024

# Credits

- These slides are based on the following documents:
- FaaS (Function-as-a-Service), IBM Cloud Education, 30 July 2019, https://www.ibm.com/cloud/learn/faas#toc-what-is-fa-j2qiBzdQ
- How are serverless computing and Platform-as-a-Service different? | PaaS vs. Serverless, Cloudflare, https://www.cloudflare.com/learning/serverless/glossary/serverless-vs-paas/
- Serverless vs. FaaS: A Beginner's Guide, February 7, 2020 https://www.liquidweb.com/kb/serverless-vs-faas-a-beginners-guide/
- Do You Need IaaS, CaaS, PaaS, or FaaS?, December 16th, 2019, https://www.trustradius.com/buyer-blog/do-you-need-iaas-caas-paas-or-faas
- Google Cloud Platform, Time to "Hello, World": VMs vs. containers vs. PaaS vs. FaaS, https://cloud.google.com/blog/products/gcp/time-to-hello-world-vms-vs-containers-vs-paas-vs-faas
- What is BaaS? | Backend-as-a-Service vs. Serverless, Cloudflare, https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/
- The Serverless Series — What Is Serverless?, What's The Difference Between IaaS, SaaS, PaaS, BaaS, FaaS, and Serverless?, Nicolas Dao, Apr 15, 2018, https://blog.neap.co/the-serverless-series-what-is-serverless-d651fbacf3f4
- Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/
- What Is Function as a Service (FaaS)?, Cloudflarehttps://www.cloudflare.com/learning/serverless/glossary/function-as-a-service-faas/
- A (Very!) Quick Comparison of Kubernetes Serverless Frameworks, Vshn Ag, 17. May 2019, https://vshn.ch/en/blog/a-very-quick-comparison-of-kubernetes-serverless-frameworks/

## Agenda

- Popularity of Serverless Computing
- Serverless Computing Versus Functions as a Service
- Serverless Computing Versus Platform as a Service
- IaaS, CaaS, PaaS and FaaS
- Serverless vs FaaS vs BaaS
- Serverless Computing versus Containers
- Functions Versus Microservices
- Serverless Computing Services
- Kubernetes Serverless Frameworks
- Comparison of Serverless Platform

## Popularity of Serverless Computing

- Use of PaaS services from public cloud providers is exploding.
- **Serverless** is the **top-growing extended cloud service** for the **second year in a row**, with a **50 percent growth over 2018** (24 to 36 percent adoption).

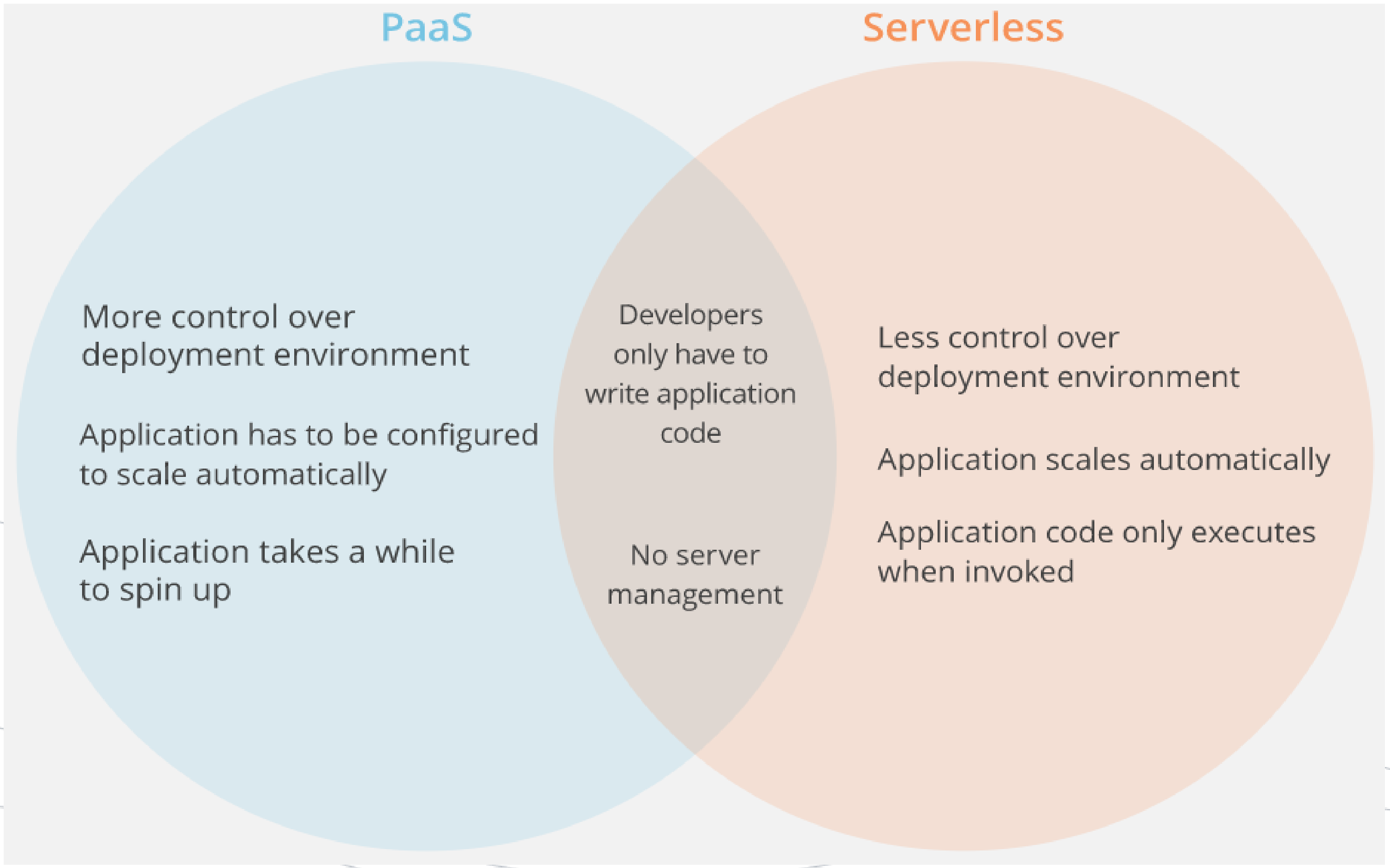
Source: 2019 RightScale® State of the Cloud Report from Flexera™

## Serverless Computing Versus Functions as a Service

- Serverless Computing and Functions-as-a-Service (FaaS) are often conflated with one another but the truth is that FaaS is actually a subset of serverless.
- Serverless Computing is focused on any service category, be it compute, storage, database, messaging, api gateways, etc. where configuration, management, and billing of servers are invisible to the end user.
- FaaS, perhaps the most central technology in serverless architectures, is focused on the event-driven computing paradigm wherein application code, or containers, only run in response to events or requests.

Source: FaaS (Function-as-a-Service), IBM Cloud Education, 30 July 2019, https://www.ibm.com/cloud/learn/faas#toc-what-is-fa-j2qiBzdQ

# Serverless Computing Versus Platform as a Service


PaaS
Serverless
More control over deployment environment
Application has to be configured to scale automatically
Application takes a while to spin up
Developers only have to write application code
No server management
Less control over deployment environment
Application scales automatically
Application code only executes when invoked

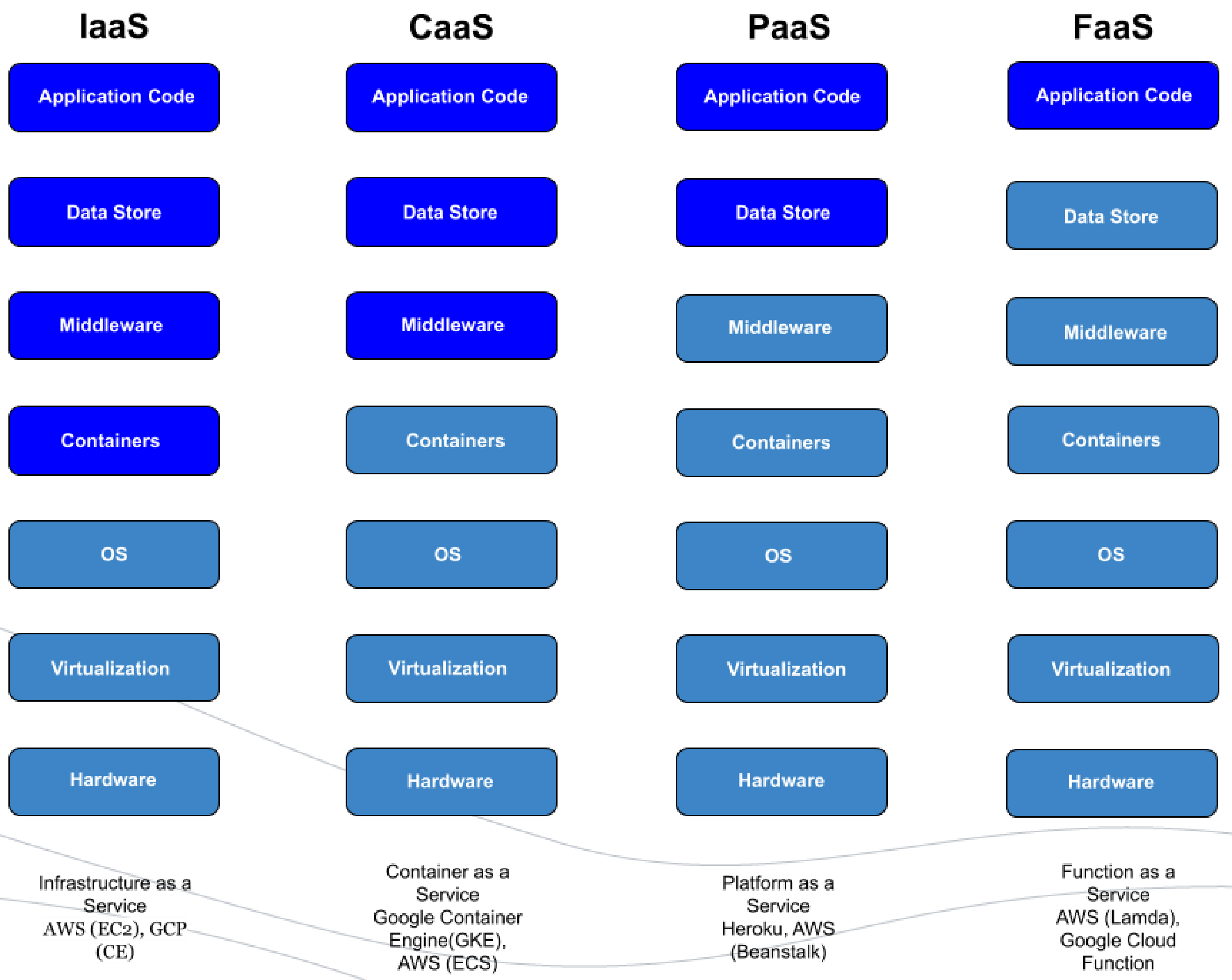
In: How are serverless computing and Platform-as-a-Service different? | PaaS vs. Serverless, Cloudflare, https://www.cloudflare.com/learning/serverless/glossary/serverless-vs-paas/

## IaaS, CaaS, PaaS and FaaS

- **IaaS** (Infrastructure as a Service), **PaaS** (Platform as a Service), **CaaS** (Containers as a Service) and **FaaS** Functions-as-a-Service, **all play a critical role** in the **serverless ecosystem**.
- FaaS is the most central and most definitional element of the serverless stack.
- **CaaS** is one step up the abstraction ladder.
- Where IaaS platforms use virtual machines or bare metal hardware as fundamental resources, CaaS platforms package up applications and all their dependencies into containers more lightweight than virtual machines.

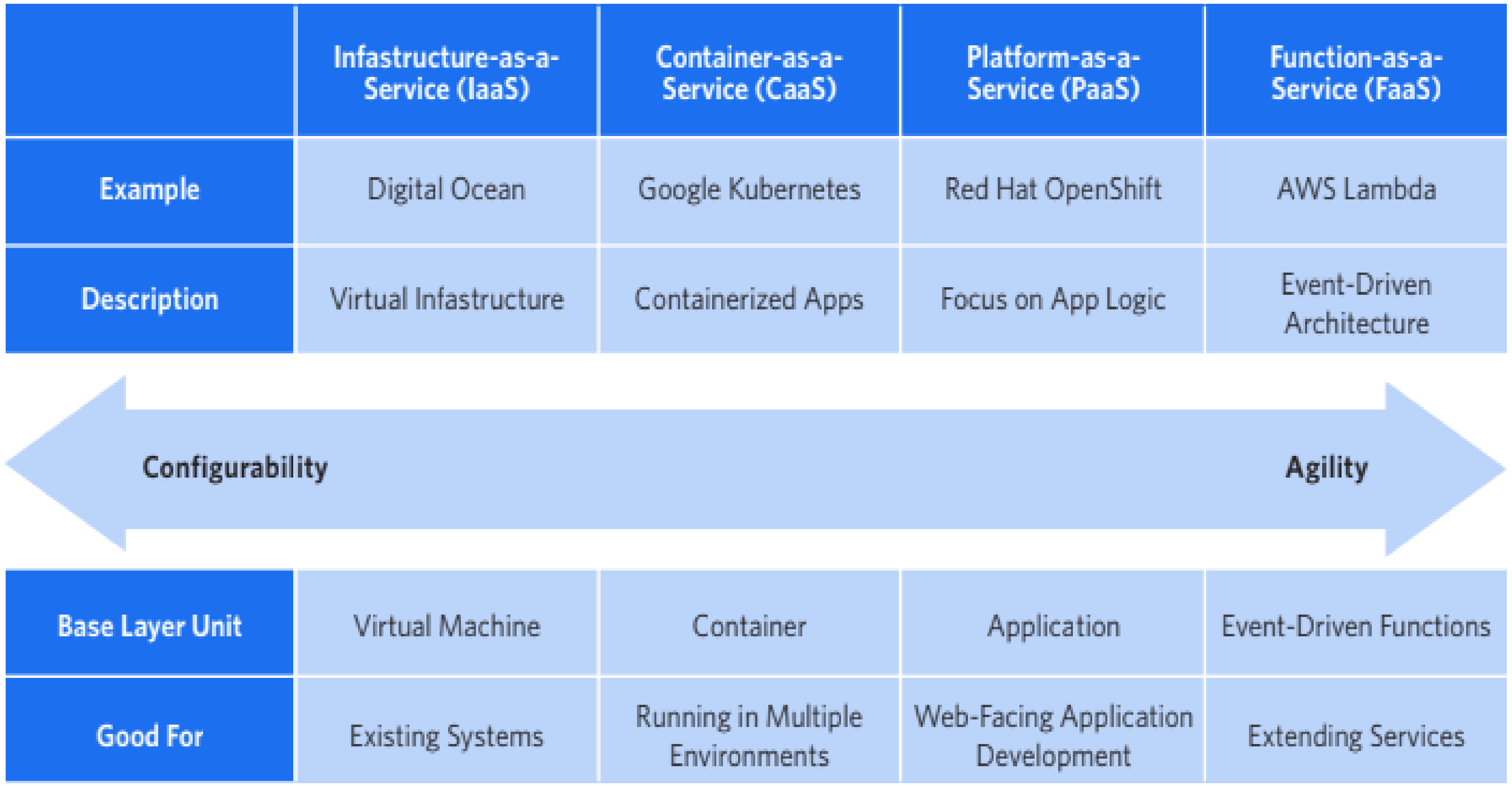
Source: FaaS (Function-as-a-Service), IBM Cloud Education, 30 July 2019, https://www.ibm.com/cloud/learn/faas#toc-what-is-fa-j2qiBzdQ

# IaaS, CaaS, PaaS and FaaS


IaaS
CaaS
PaaS
FaaS
Application Code
Data Store
Middleware
Containers
OS
Virtualization
Hardware
Infrastructure as a Service AWS (EC2), GCP (CE)
Container as a Service Google Container Engine(GKE), AWS (ECS)
Platform as a Service Heroku, AWS (Beanstalk)
Function as a Service AWS (Lamda), Google Cloud Function

In: Serverless vs. FaaS: A Beginner's Guide, February 7, 2020 https://www.liquidweb.com/kb/serverless-vs-faas-a-beginners-guide/

# IaaS, CaaS, PaaS and FaaS



| | Infastructure-as-a-Service (IaaS) | Container-as-a-Service (CaaS) | Platform-as-a-Service (PaaS) | Function-as-a-Service (FaaS) |
|---|---|---|---|---|
| Example | Digital Ocean | Google Kubernetes | Red Hat OpenShift | AWS Lambda |
| Description | Virtual Infastructure | Containerized Apps | Focus on App Logic | Event-Driven Architecture |

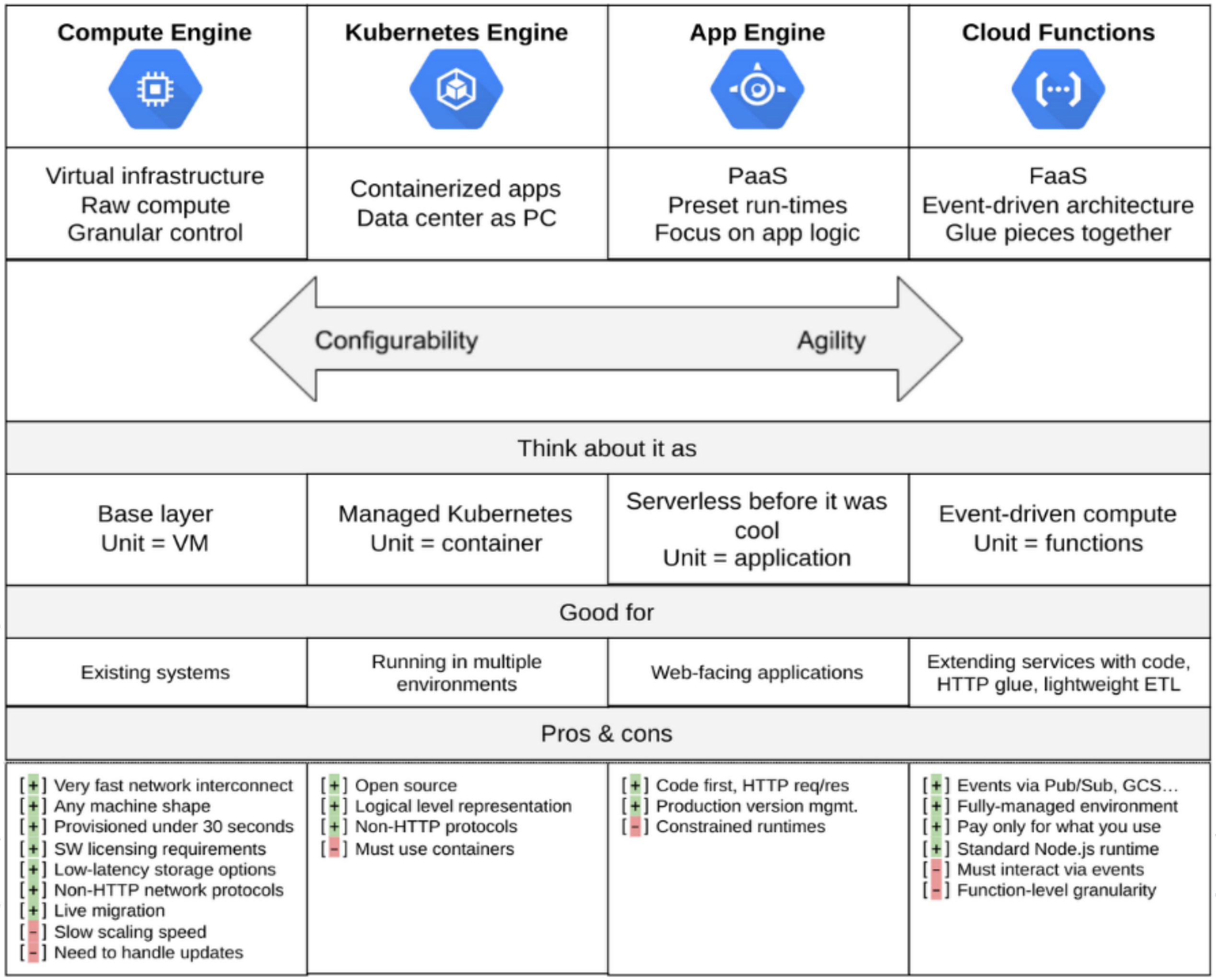
Configurability
Agility

| | | | | |
|---|---|---|---|---|
| Base Layer Unit | Virtual Machine | Container | Application | Event-Driven Functions |
| Good For | Existing Systems | Running in Multiple Environments | Web-Facing Application Development | Extending Services |

**This is a modified version of Google computing stack illustration*

In: Do You Need IaaS, CaaS, PaaS, or FaaS?, December 16th, 2019, https://www.trustradius.com/buyer-blog/do-you-need-iaas-caas-paas-or-faas

# IaaS, CaaS, PaaS and FaaS



| Compute Engine | Kubernetes Engine | App Engine | Cloud Functions |
|---|---|---|---|
| Virtual infrastructure<br>Raw compute<br>Granular control | Containerized apps<br>Data center as PC | PaaS<br>Preset run-times<br>Focus on app logic | FaaS<br>Event-driven architecture<br>Glue pieces together |
| Configurability | | | Agility |
| Think about it as | | | |
| Base layer<br>Unit = VM | Managed Kubernetes<br>Unit = container | Serverless before it was cool<br>Unit = application | Event-driven compute<br>Unit = functions |
| Good for | | | |
| Existing systems | Running in multiple environments | Web-facing applications | Extending services with code, HTTP glue, lightweight ETL |
| Pros & cons | | | |
| [+] Very fast network interconnect<br>[+] Any machine shape<br>[+] Provisioned under 30 seconds<br>[+] SW licensing requirements<br>[+] Low-latency storage options<br>[+] Non-HTTP network protocols<br>[+] Live migration<br>[-] Slow scaling speed<br>[-] Need to handle updates | [+] Open source<br>[+] Logical level representation<br>[+] Non-HTTP protocols<br>[-] Must use containers | [+] Code first, HTTP req/res<br>[+] Production version mgmt.<br>[-] Constrained runtimes | [+] Events via Pub/Sub, GCS…<br>[+] Fully-managed environment<br>[+] Pay only for what you use<br>[+] Standard Node.js runtime<br>[-] Must interact via events<br>[-] Function-level granularity |

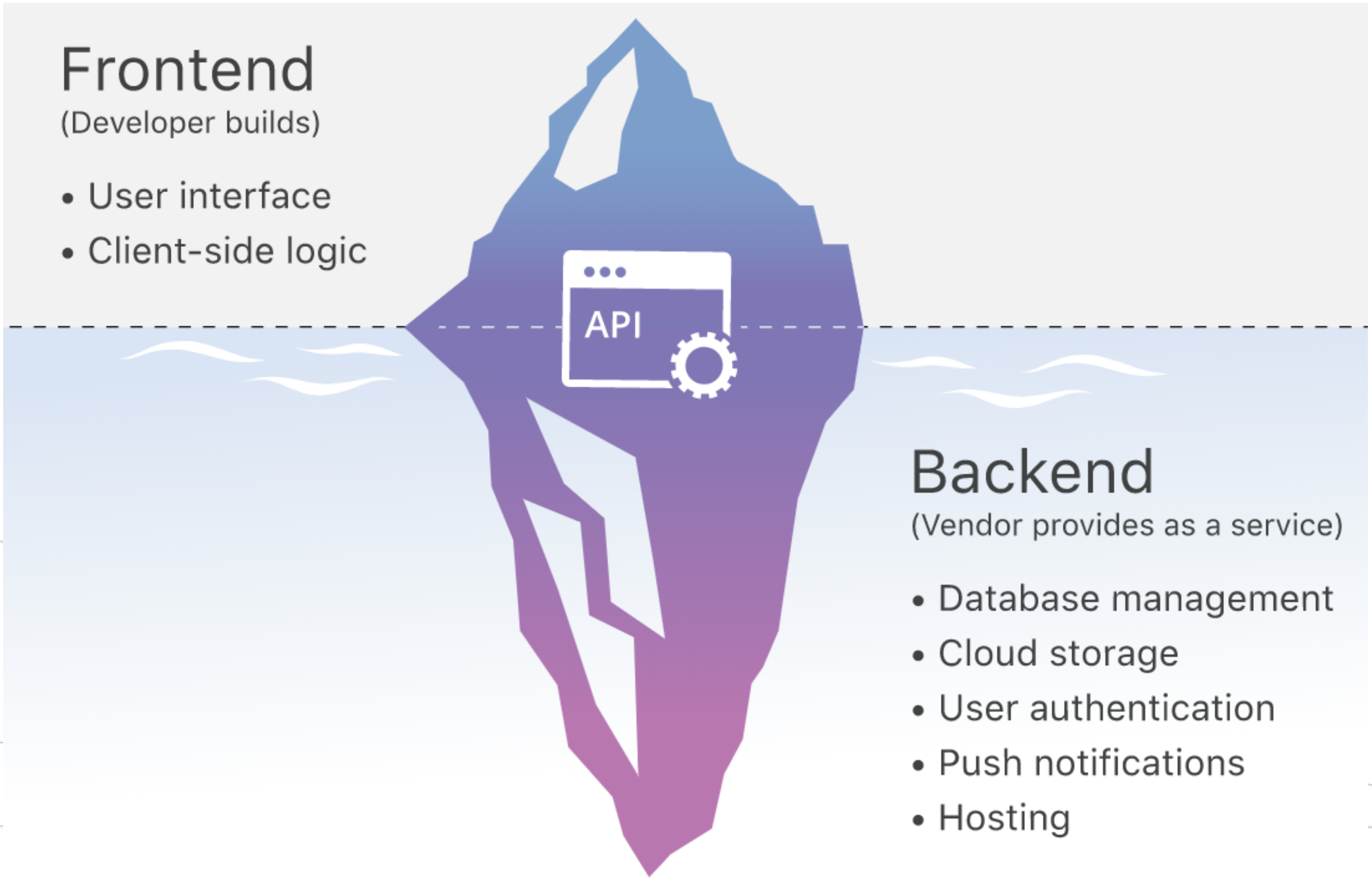
In: Google Cloud Platform, Time to “Hello, World”: VMs vs. containers vs. PaaS vs. FaaS, https://cloud.google.com/blog/products/gcp/time-to-hello-world-vms-vs-containers-vs-paas-vs-faas

## Serveless vs FaaS vs BaaS

- **Backend-as-a-Service (BaaS)** is a cloud service model in which developers outsource all the behind-the-scenes aspects of a web or mobile application so that they only have to write and maintain the frontend.
- **BaaS vendors provide pre-written software** for activities that take place on servers, such as user authentication, database management, remote updating, and push notifications (for mobile apps), as well as cloud storage and hosting.

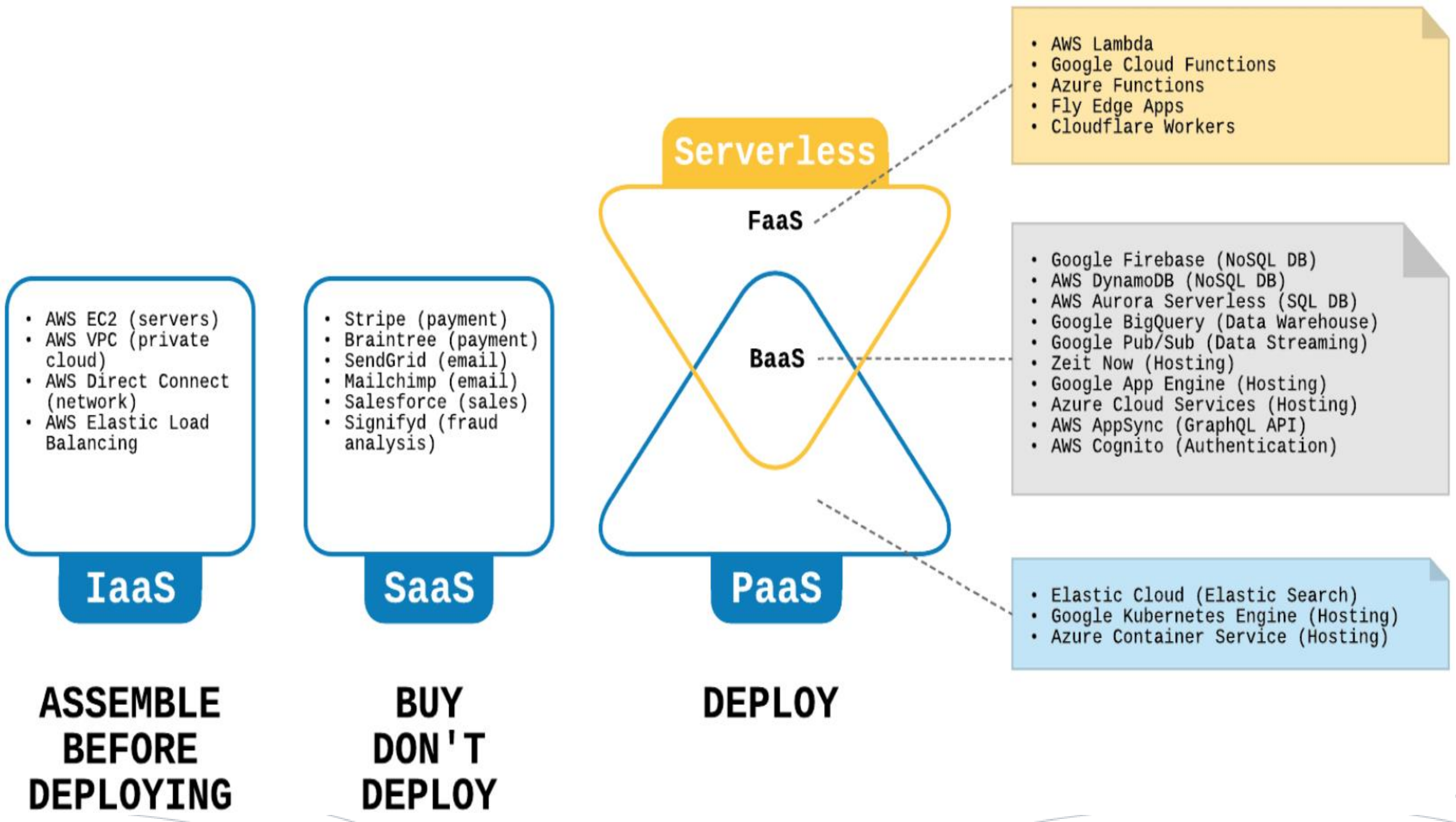
Source: What is BaaS? | Backend-as-a-Service vs. Serverless, Cloudflare, https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/

## Serveless vs FaaS vs BaaS



In: What is BaaS? | Backend-as-a-Service vs. Serverless, Cloudflare, https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/

## Serveless vs FaaS vs BaaS



Serverless
FaaS
BaaS
IaaS
SaaS
PaaS
• AWS EC2 (servers)
• AWS VPC (private cloud)
• AWS Direct Connect (network)
• AWS Elastic Load Balancing
• Stripe (payment)
• Braintree (payment)
• SendGrid (email)
• Mailchimp (email)
• Salesforce (sales)
• Signifyd (fraud analysis)
• AWS Lambda
• Google Cloud Functions
• Azure Functions
• Fly Edge Apps
• Cloudflare Workers
• Google Firebase (NoSQL DB)
• AWS DynamoDB (NoSQL DB)
• AWS Aurora Serverless (SQL DB)
• Google BigQuery (Data Warehouse)
• Google Pub/Sub (Data Streaming)
• Zeit Now (Hosting)
• Google App Engine (Hosting)
• Azure Cloud Services (Hosting)
• AWS AppSync (GraphQL API)
• AWS Cognito (Authentication)
• Elastic Cloud (Elastic Search)
• Google Kubernetes Engine (Hosting)
• Azure Container Service (Hosting)
ASSEMBLE BEFORE DEPLOYING
BUY DON'T DEPLOY
DEPLOY

In: The Serverless Series — What Is Serverless?, What's The Difference Between IaaS, SaaS, PaaS, BaaS, FaaS, and Serverless?, Nicolas Dao, Apr 15, 2018, https://blog.neap.co/the-serverless-series-what-is-serverless-d651fbacf3f4

## Serverless Computing versus Containers

- Serverless computing and containers are both architectures that reduce overhead for cloud-hosted web applications, but they differ in several important ways.
- Containers are more lightweight than virtual machines, but serverless deployments are even more lightweight and scale more easily than container-based architectures.

Source: Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/

## Serverless Computing versus Containers

- Both serverless computing and containers enable developers to build applications with far less overhead and more flexibility than applications hosted on traditional servers or virtual machines.

- Which style of architecture a developer should use depends on the needs of the application, but serverless applications are more scalable and usually more cost-effective.

Source: Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/

## Serverless Computing versus Containers

- A container 'contains' both an application and all the elements the application needs to run properly, including system libraries, system settings, and other dependencies.

- Like a 'just add water' pancake mix, containers only need one thing – to be hosted and run – in order to perform their function.

Source: Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/

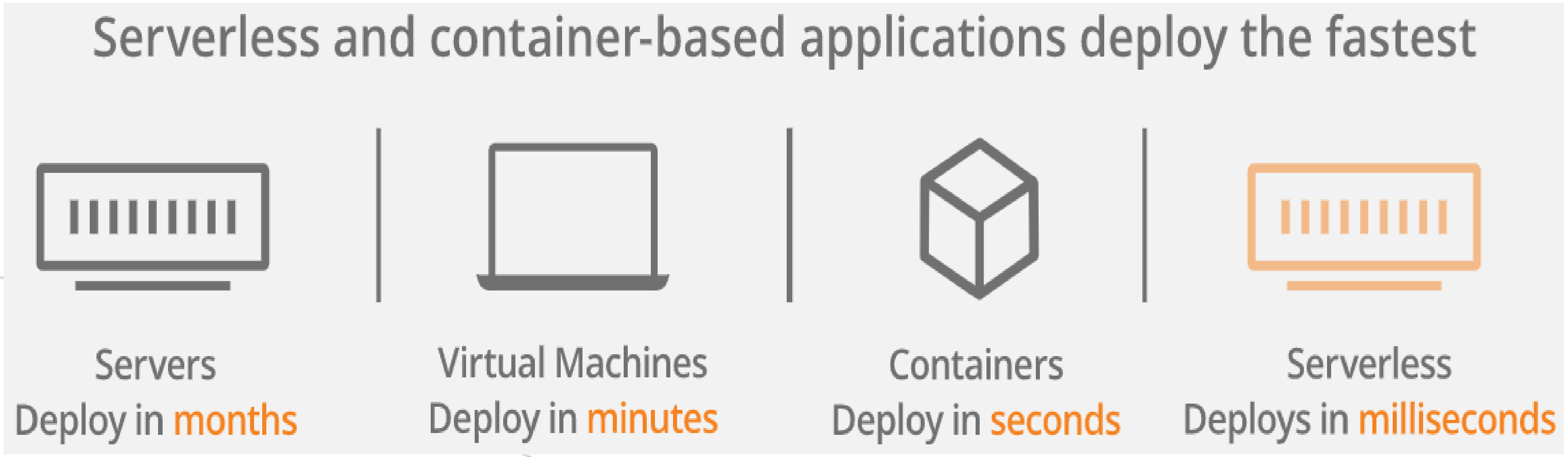
## Serverless Computing versus Containers

- **Time of Deployment**
- Containers take longer to set up initially than serverless functions because it is necessary to configure system settings, libraries, and so on.
- Once configured, containers take only a few seconds to deploy.
- But because serverless functions are smaller than container microservices and do not come bundled with system dependencies, they only take milliseconds to deploy.
- Serverless applications can be live as soon as the code is uploaded.

Source: Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/

# Serverless Computing versus Containers

## Time of Deployment from Servers to Serverless



In: Serverless Computing vs. Containers | How to Choose, Cloudflare, https://www.cloudflare.com/learning/serverless/serverless-vs-containers/?utm_referrer=https://www.google.com/

# Functions Versus Microservices

- **Function as a service** (FaaS) is a serverless way to execute modular pieces of code on the edge. FaaS lets developers write and update a piece of code on the fly, which can then be executed in response to an event, such as a user clicking on an element in a web application. This makes it easy to scale code and is a cost-efficient way to implement microservices.
- **Microservices**: If a web application were a work of visual art, using microservice architecture would be like making the art out of a collection of mosaic tiles. The artist can easily add, replace, and repair one tile at a time. Monolithic architecture would be like painting the entire work on a single piece of canvas.

Source: What Is Function as a Service (FaaS)?, Cloudflarehttps://www.cloudflare.com/learning/serverless/glossary/function-as-a-service-faas/

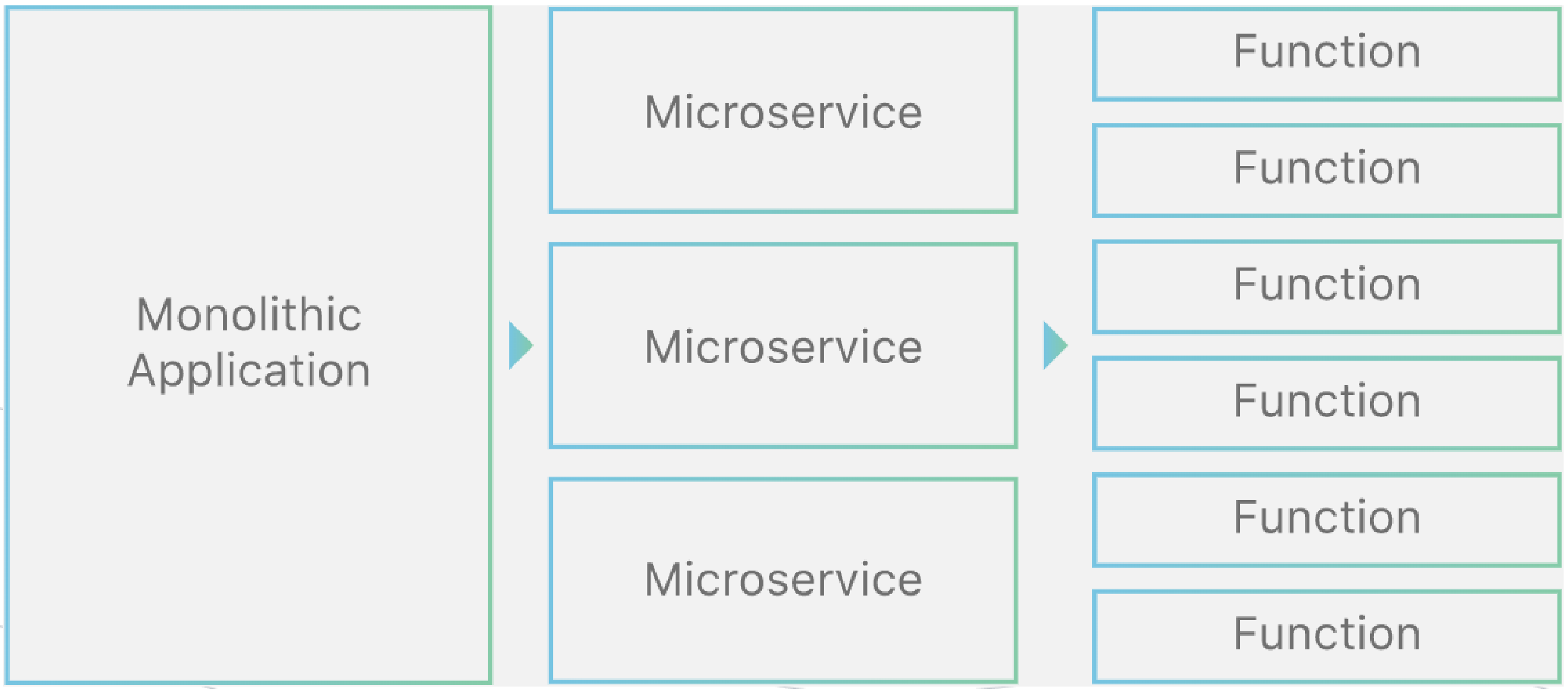
## Functions Versus Microservices

- This approach of building an application out of a set of modular components is known as **microservice architecture**.
- **Dividing** an **application into microservices** is appealing to developers because it means they **can create** and **modify small pieces of code** which can be easily implemented into their codebases.
- This is in contrast to monolithic architecture, in which all the code is interwoven into one large system. With large monolithic systems, even a minor changes to the application requires a hefty deploy process. FaaS eliminates this deploy complexity.
- Using serverless code like FaaS, web developers can focus on writing application code, while the serverless provider takes care of server allocation and backend services.

Source: What Is Function as a Service (FaaS)?, Cloudflarehttps://www.cloudflare.com/learning/serverless/glossary/function-as-a-service-faas/

# Functions Versus Microservices

- **Function as a service Versus microservices (3)**



In: What Is Function as a Service (FaaS)?, Cloudflarehttps://www.cloudflare.com/learning/serverless/glossary/function-as-a-service-faas/

## Serverless Computing Services

- **AWS Lambda** (formerly AWS Lambda Functions): https://aws.amazon.com/pt/lambda/.
  Allows to run codes without thinking about servers. Customer only pays for the computing time they use.
- **Azure Functions**:
  https://azure.microsoft.com/pt-pt/services/functions/
  Allows to build applications faster with a serverless architecture. Users may accelerate development with an event-oriented, serverless computing experience. Scale on demand and pay only for the consumed resources.

## Serverless Computing Services

- **Google Cloud Functions**: https://cloud.google.com/functions/
Event-based serverless computing platform. It does automatic escalation, is highly available and fault tolerant. Users pay only when their code is running.
- **Cloud Functions for Firebase**: https://firebase.google.com/products/functions/
Users may run mobile backend code without managing servers.
- **Apache OpenWhisk**
https://openwhisk.apache.org/
Apache OpenWhisk (Incubating) is an open source, distributed Serverless platform that executes functions (fx) in response to events at any scale.

# Serverless Computing Services

- **OpenWhisk** manages the infrastructure, servers and scaling using Docker containers.
- Since Apache OpenWhisk builds its components using containers it easily supports many deployment options both locally and within cloud infrastructures. Options include many of today's popular Container frameworks such as Kubernetes, Mesos, OpenShift and Compose.
- In general, the community endorses deployment on Kubernetes using Helm charts since it provides many easy and convenient implementations for both Developers and Operators alike such as Minikube.

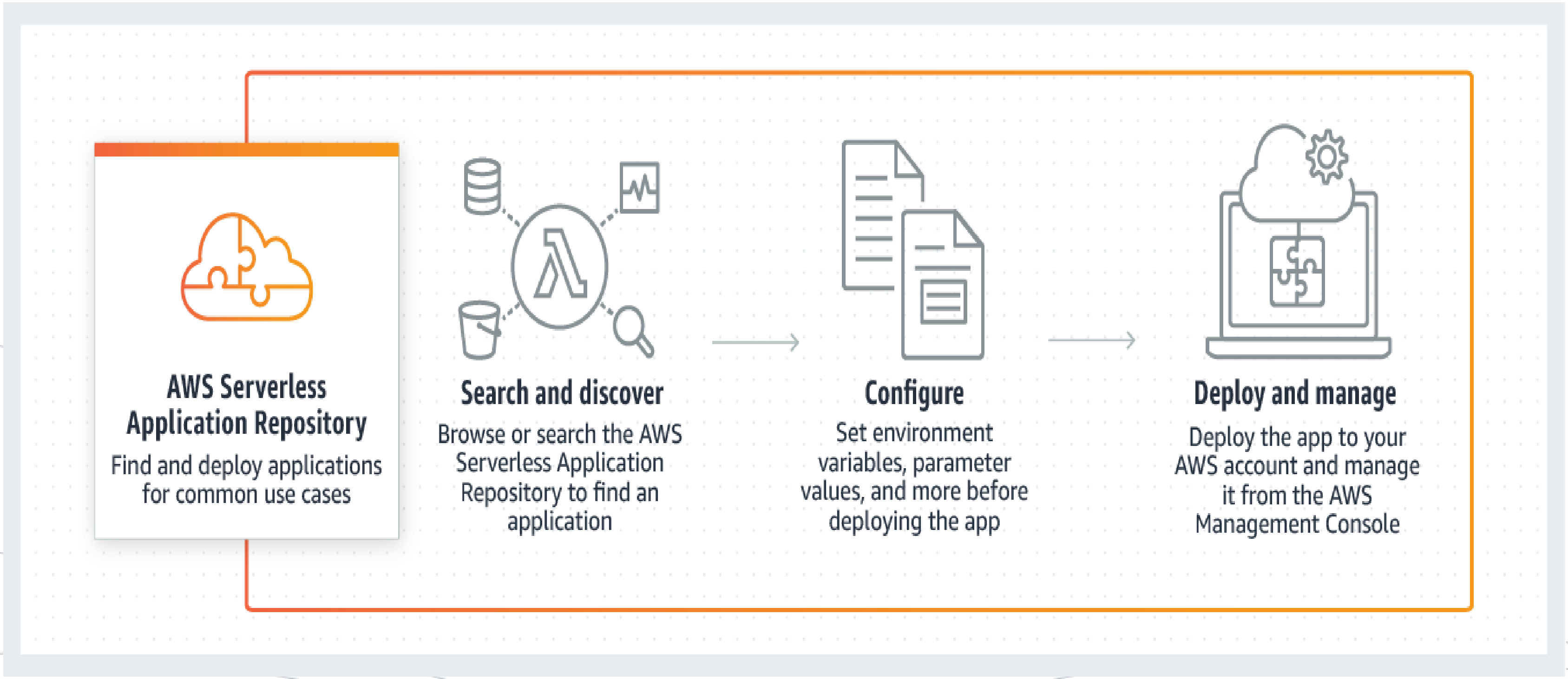
Source: Apache OpenWhisk, Open Source Serverless Cloud Platform, https://openwhisk.apache.org/

## Serverless Computing Services

- **AWS Serverless Application Repository** (https://aws.amazon.com/pt/serverless/serverlessrepo/)
- The AWS Serverless Application Repository is a managed repository for serverless applications.
- Allows to use **pre-built applications** of the Serverless Application Repository in the user serverless architecture.
- Each application is packaged with an AWS Serverless Application Model (SAM) template that defines the AWS features used.
- There is no additional cost for using the serverless application repository; Users only pay for the AWS features used in the applications they deploy.

## Serverless Computing Services

### AWS Serverless Application Repository: How It Works - Application Deployment


AWS Serverless Application Repository
Find and deploy applications for common use cases
Search and discover
Browse or search the AWS Serverless Application Repository to find an application
Configure
Set environment variables, parameter values, and more before deploying the app
Deploy and manage
Deploy the app to your AWS account and manage it from the AWS Management Console

In: AWS Serverless Application Repository, https://aws.amazon.com/pt/serverless/serverlessrepo/

## Kubernetes Serverless Frameworks

- **Kubernetes** is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
- **Kubernetes** has a large, rapidly growing ecosystem.
- **Serverless Frameworks for Kubernetes**:
  - OpenFaas (https://www.openfaas.com/)
  - Fn Project (https://fnproject.io/)
  - Fission (https://fission.io/)
  - OpenWhisk (https://openwhisk.apache.org/)
  - Kubeless (https://kubeless.io/)
  - TriggerMesh (https://triggermesh.com/)
  - Knative (https://knative.dev/)

Source: What is Kubernetes?, Kubernetes, https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Source: A (Very!) Quick Comparison of Kubernetes Serverless Frameworks, Vshn Ag, 17. May 2019, https://vshn.ch/en/blog/a-very-quick-comparison-of-kubernetes-serverless-frameworks/

# Comparison of Serverless Platform

- **Comparison of Serverless Platforms**
- Comparison of Serverless Function (FaaS) Providers
  https://fauna.com/blog/comparison-faas-providers
- Comparing Serverless Architecture Providers: AWS, Azure, Google, IBM, and other FaaS vendors
  https://www.altexsoft.com/blog/cloud/comparing-serverless-architecture-providers-aws-azure-google-ibm-and-other-faas-vendors/
- A (Very!) Quick Comparison of **Kubernetes Serverless** Frameworks
  https://vshn.ch/en/blog/a-very-quick-comparison-of-kubernetes-serverless-frameworks/
- A Comparison of **Serverless Frameworks for Kubernetes**: OpenFaas, OpenWhisk, Fission, Kubeless and more
  https://winderresearch.com/a-comparison-of-serverless-frameworks-for-kubernetes-openfaas-openwhisk-fission-kubeless-and-more/