



AU PROGRAMME

[ CLOUD DEVOPS]

Serverless avec Google Cloud

L'univers de l'hébergement Cloud est très vaste et nécessite de s'approprier de nouveaux paradigmes afin d'exploiter au maximum ces infrastructures.

Le Serverless est une solution Cloud qui permet de rapidement mettre en production vos applications : le serveur et le runtime sont gérés directement par la solution permettant aux équipes de développement de se concentrer sur le code métier.

Que ce soit avec Google App Engine, la solution de PaaS (Platform as a Service) ou encore avec les Cloud Functions, la solution FaaS (Function As a Service) ou encore Cloud Run, Google Cloud fournit un environnement et des outils pour être productif dans le Cloud.

Venez découvrir ces différentes solutions !



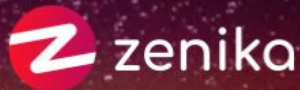
Julien Landuré

Directeur Technique



Eric Briand

Directeur Technique



24 mars 2022

Julien Landuré



Directeur technique Groupe



GDE Cloud + Official Trainer

Orga GDG Cloud Nantes / DevFest



@jlandure



@jlandure



Éric Briand



Directeur technique Nantes



GDE Cloud + Official Trainer

Orga CNCF Meetup Nantes



@eric_briand



@ebriand



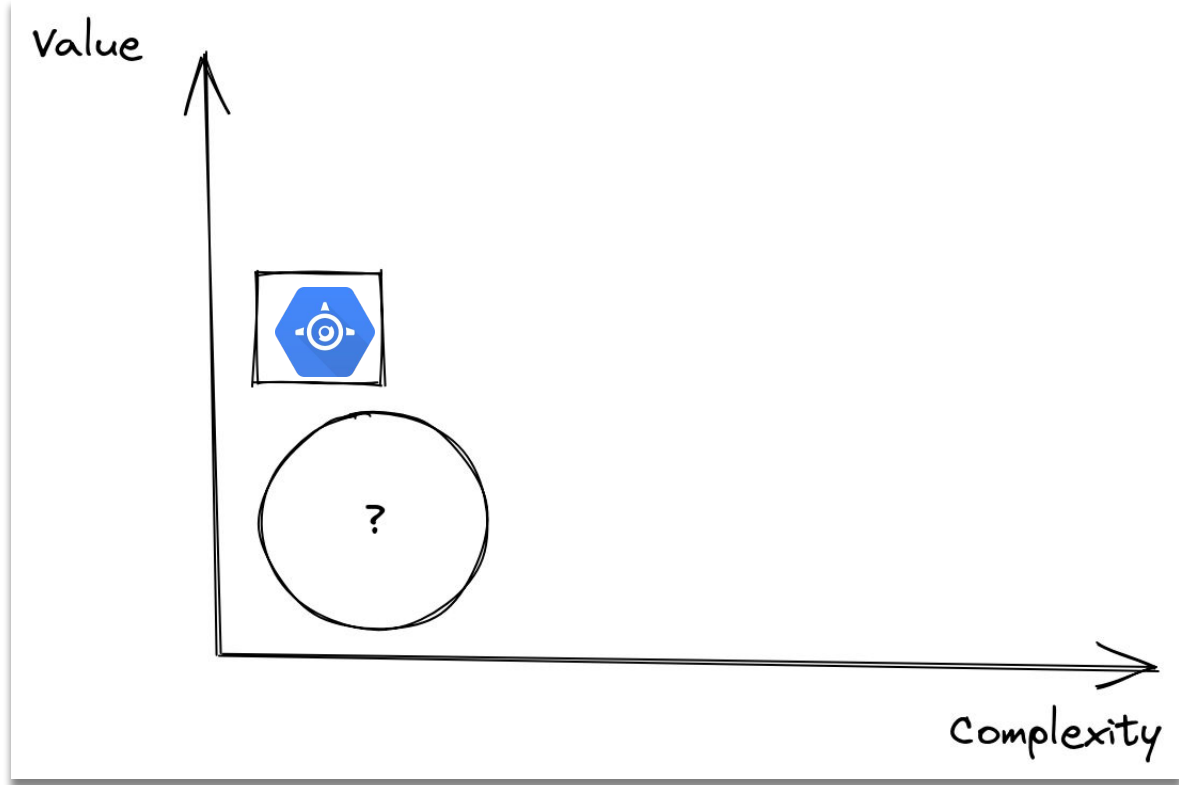
Back to the future...



App Engine in 2010



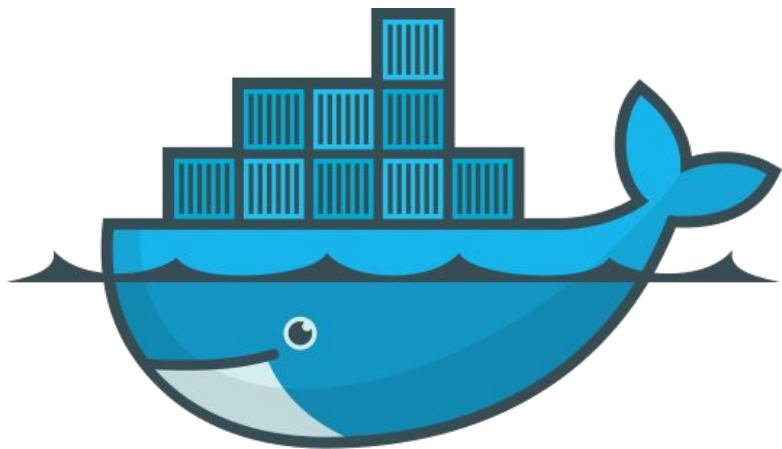
Value x Complexity



Containerization

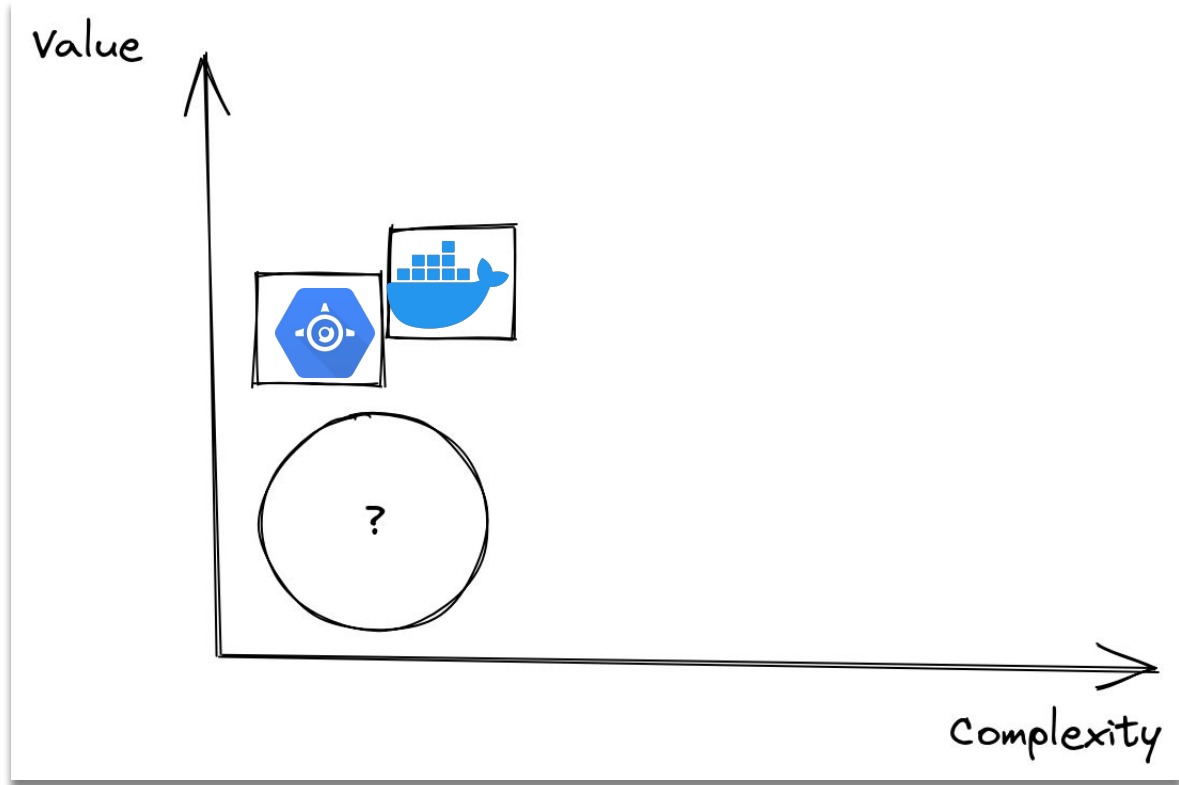


Docker in 2016



docker

Value x Complexity



Orchestration

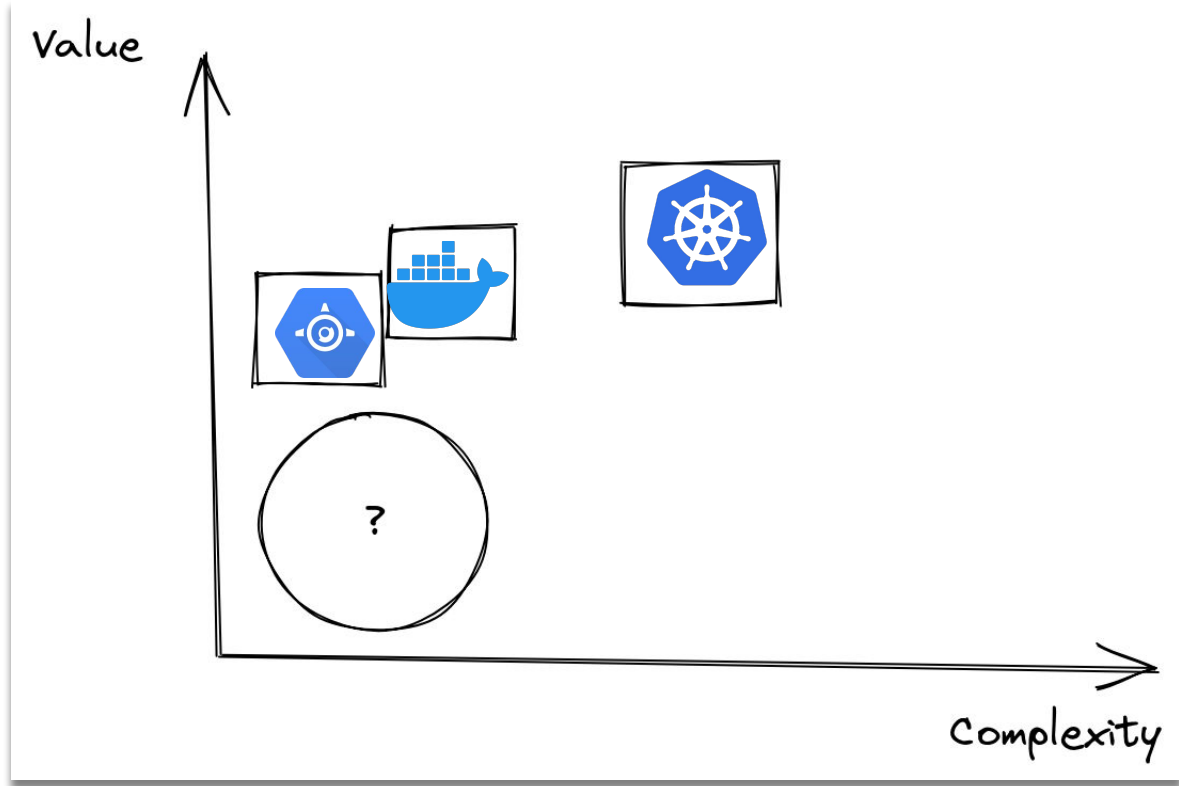


Kubernetes in 2018



kubernetes

Value x Complexity

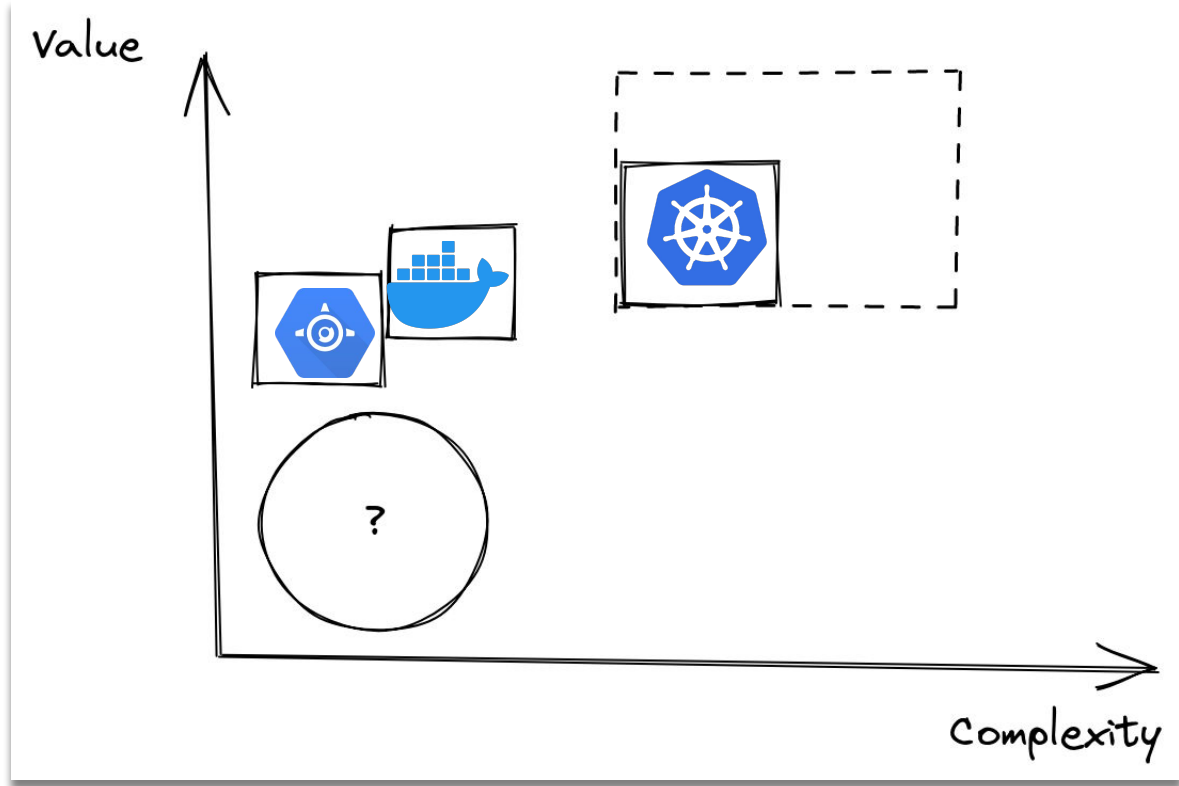


Wait!

Kubernetes can do so much...



Value x Complexity



Serverless

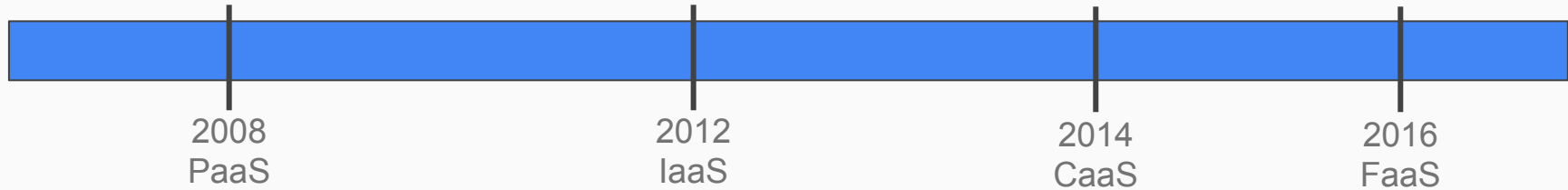


Timeline *aas

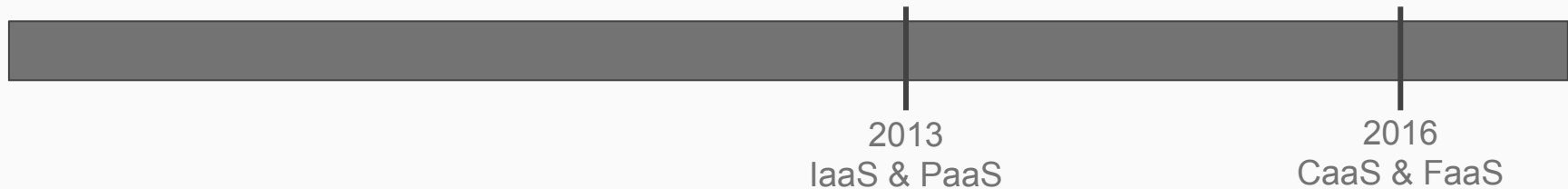
AWS



Google Cloud



Azure



Serverless, just a buzzword? 🤔



<https://youtu.be/P8pCWhPI9B4>

Cloud Run in 2019

Cloud Functions



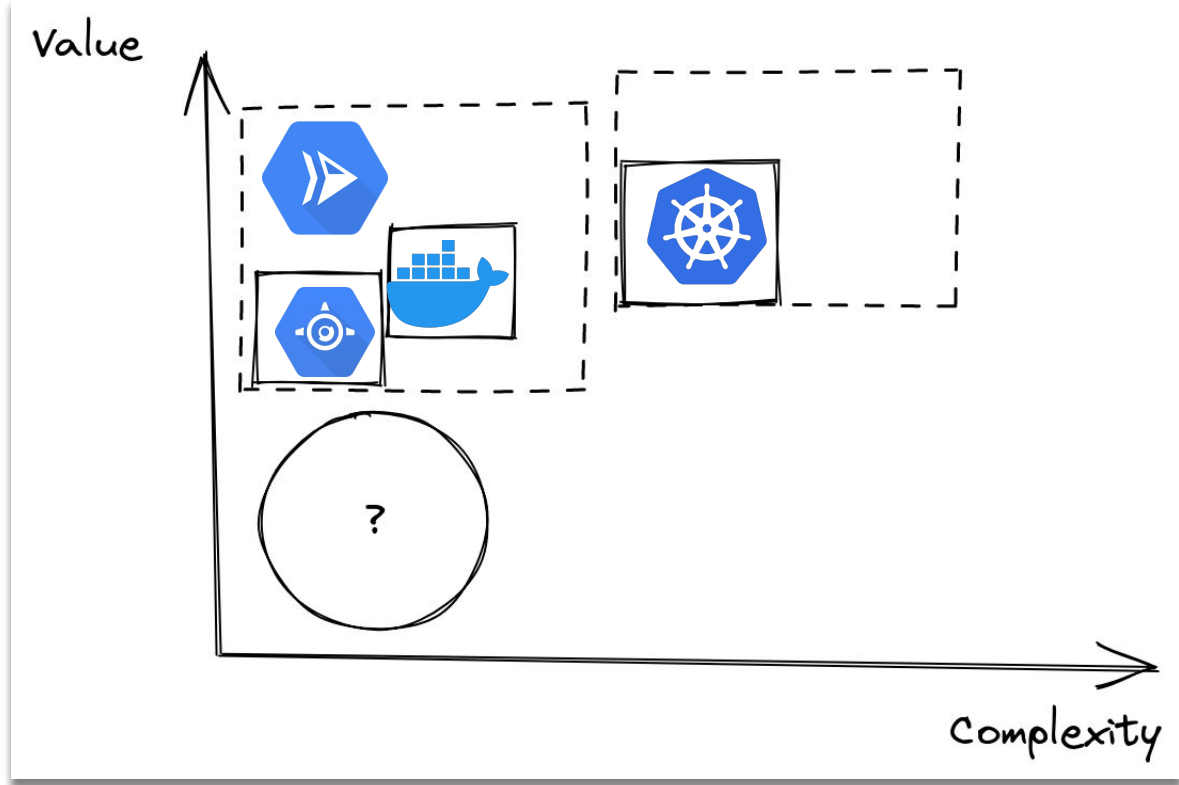
App Engine



Cloud Run



Value x Complexity

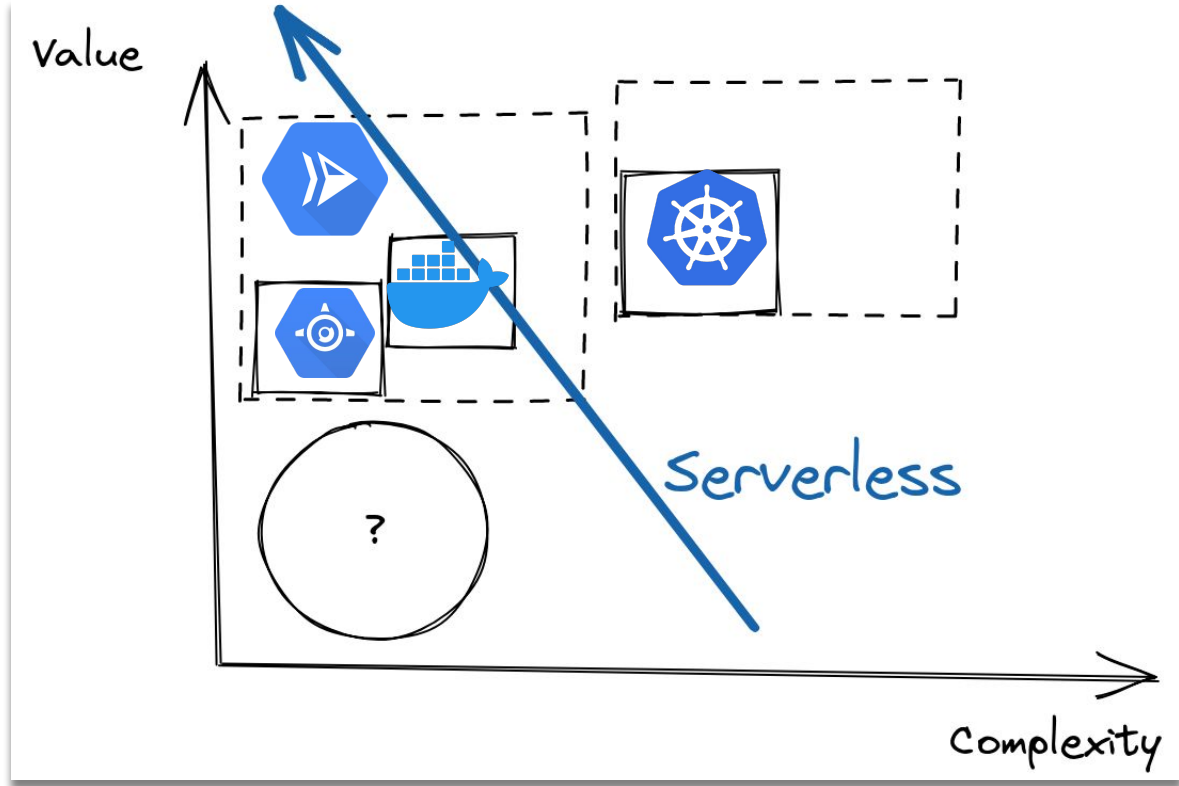


Serverless definition

Serverless computing refers to the concept of building and running applications that **do not require server management**. It describes a finer-grained deployment model where **applications, bundled as one or more functions**, are uploaded to a platform and then **executed, scaled, and billed in response to the exact demand needed at the moment**.

<https://github.com/cncf/wg-serverless/tree/master/whitepapers/serverless-overview>

Value x Complexity



The Serverless spectrum on Google Cloud



Serverless

Operational Model



No Infra Management



Managed Security



Pay only for usage

Programming Model



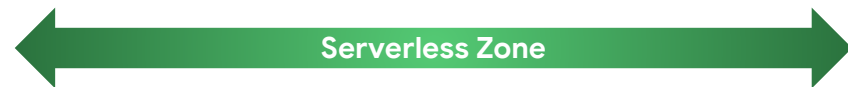
Service-based



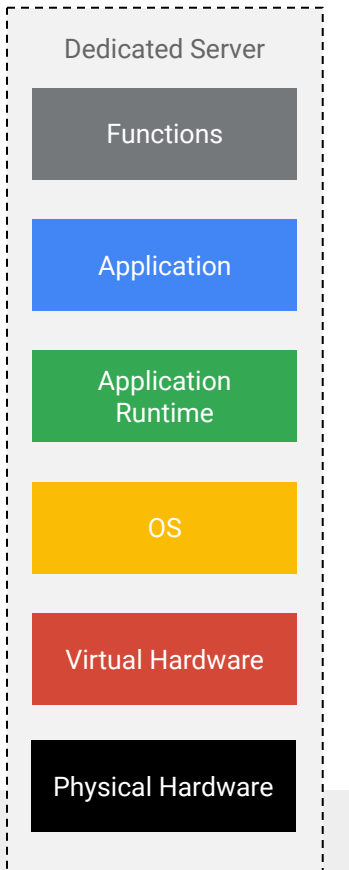
Event-driven



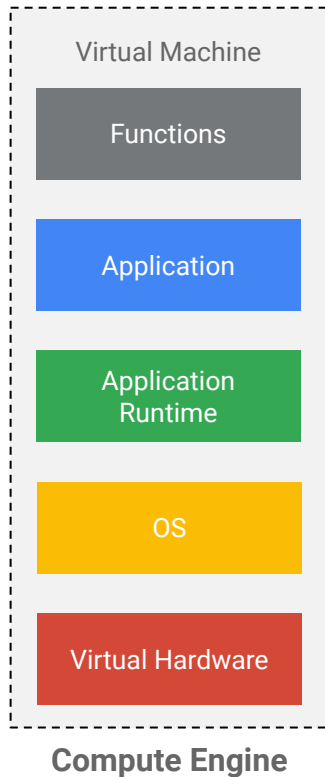
Stateless



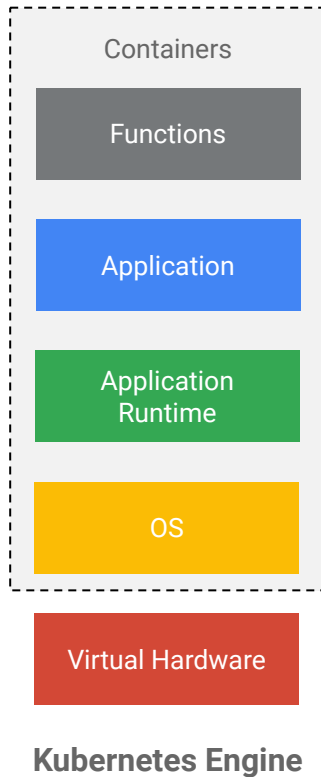
On-Premises



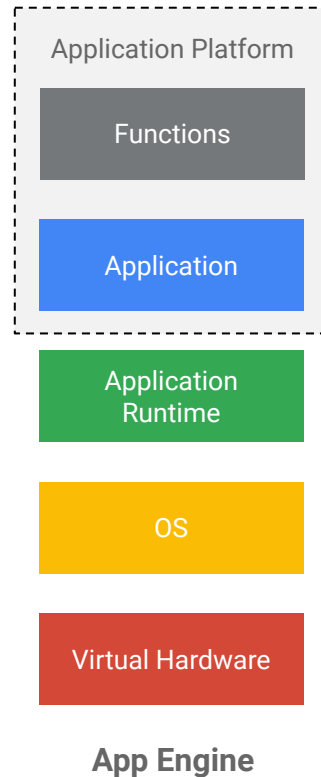
Infrastructure



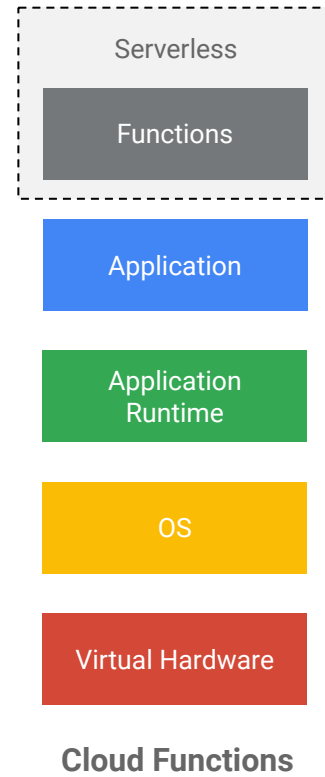
Container



Platform



Function



Cloud Run

Serverless compute options

Serverless
functions



Cloud
Functions



Source code-based event
driven functions

Serverless
web applications



App Engine



Source code-based web
apps and API backends

Serverless
HTTP workloads



Cloud Run



HTTP containers
fully managed

HTTP containers
on GKE cluster



Cloud Run



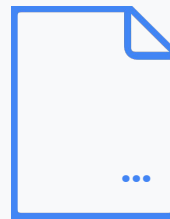
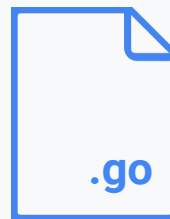
Containers

Any language

Any library

Any binary

Ecosystem of
base images

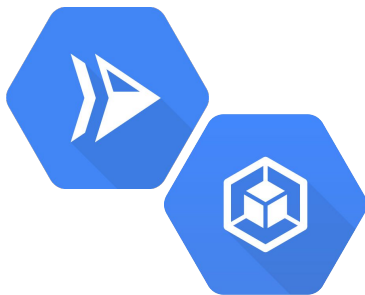


Serverless containers with Knative & Cloud Run



Cloud Run

Fully managed, deploy your workloads and don't see the cluster.



Cloud Run on Anthos

Deploy into Anthos, run serverless side-by-side with your existing workloads.



Knative Everywhere

Use the same APIs and tooling anywhere you run Kubernetes with Knative.

Container contract



Listen on 0.0.0.0 on port \$PORT (*default 8080*)

HTTP server must start < 4 min (*timeout → 504*)

Request time < 60 min (*default → 5 min*)

Stateless (*in-memory file system, doesn't persist*)

Computation only within request (*No background activity*)

Container resources

1 vCPU per container instance *(configurable to 4vCPU)*

512 MiB of memory up to a max of 8 GiB *(configurable)*

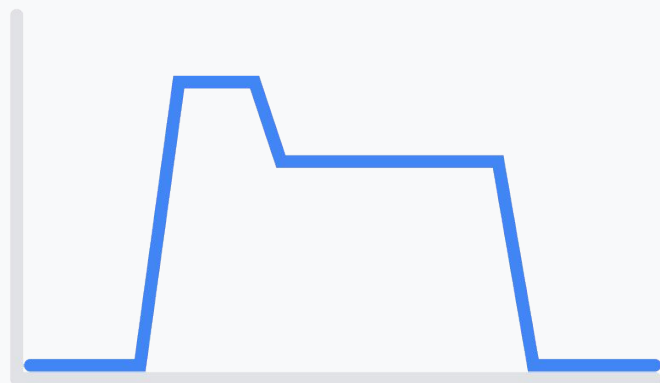
80 concurrent requests per container *(configurable 1-1000)*

1000 max containers by default *(configurable 1-1000)*

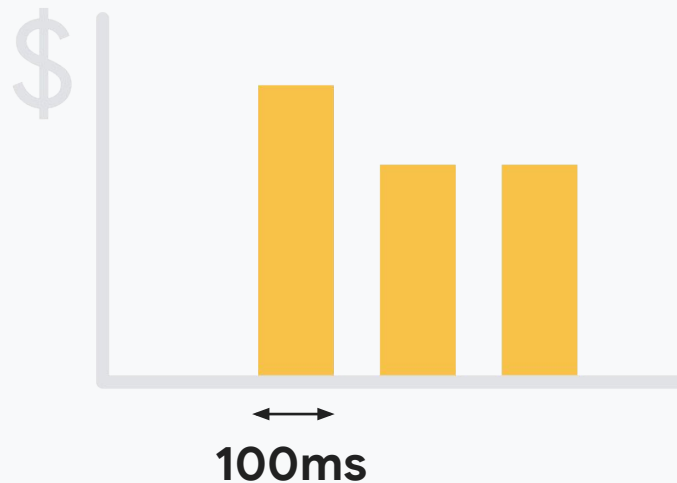
Access to a Metadata Server

Sandboxed by **gVisor**

Pay per use



CPU / Memory / Requests



100ms

Good fit? Constraints?



Good fit

Stateless

HTTP request-response workloads

Scale: way up, down to zero, bursty

Any runtime: any language, any

dependencies, etc

Constraints

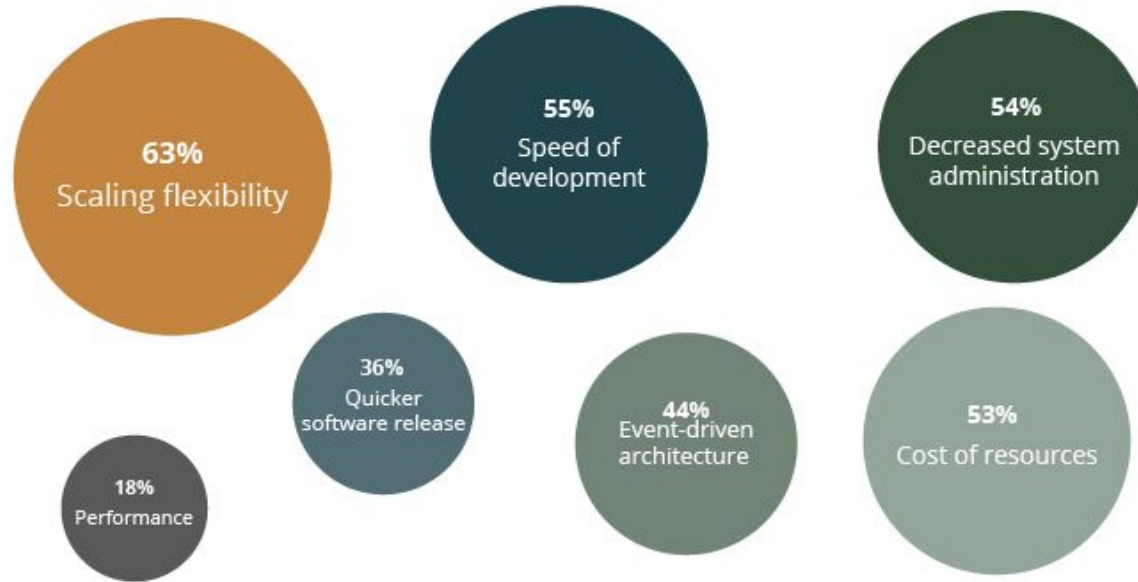
Must use containers

No background tasks

Demo 🙌



Serverless trends



State of Serverless 2020 report



Our point of view

- Serverless solutions cover the majority of app use cases
- Serverless can be interoperable
- Serverless enable a better Dev Experience (local emulator & buildpacks)
- Serverless container solutions are the future
 - **Cloud Run** is one of the TOP project Google Cloud
 - Release of AWS App Runner
 - Preview of Azure Container Instance *PREVIEW*



Serverless Everywhere

- 👍 Un sujet tendance avec de vrais concepts
- 💪 Un usage “Serverless” large
 - PubSub & Messaging
 - [Cloud SQL & Serverless exports](#)
 - Google BigQuery “serverless data warehouse”
 - AutoML & IA services
- 💰 Démarche FinOps : des tarifs avantageux !
- 🔒 Moins de choses à gérer ➡ meilleure sécurité



Merci ! Des questions ?

Formations Zenika à propos du sujet

- <https://training.zenika.com/fr-fr/officials/gcp>
- [Google Cloud Fundamentals: Core Infrastructure \(officielle\)](#)
- [Séminaire Cloud : l'état de l'art](#)



@jlandure



@eric_briand