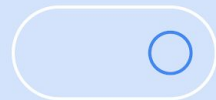
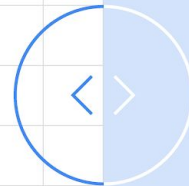
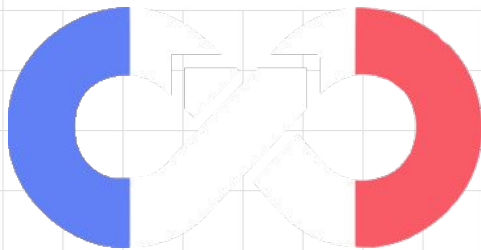




Cloud Run... everything you need without K8S

France DevOps





Julien Landuré

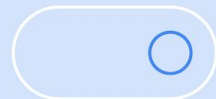
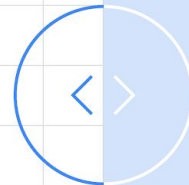
 @jlandure jlandure.dev

Google Developer Expert - Cloud

CTO Group - Zenika

GDG Cloud Nantes & DevFest Organizer

devfest.gdgnantes.com



Back to the future...

App Engine in 2010



Google
App Engine

App Engine

One of the first PaaS

Limited runtimes

Need specific libraries

Vendor lock in

Easy deployment

Custom domain name

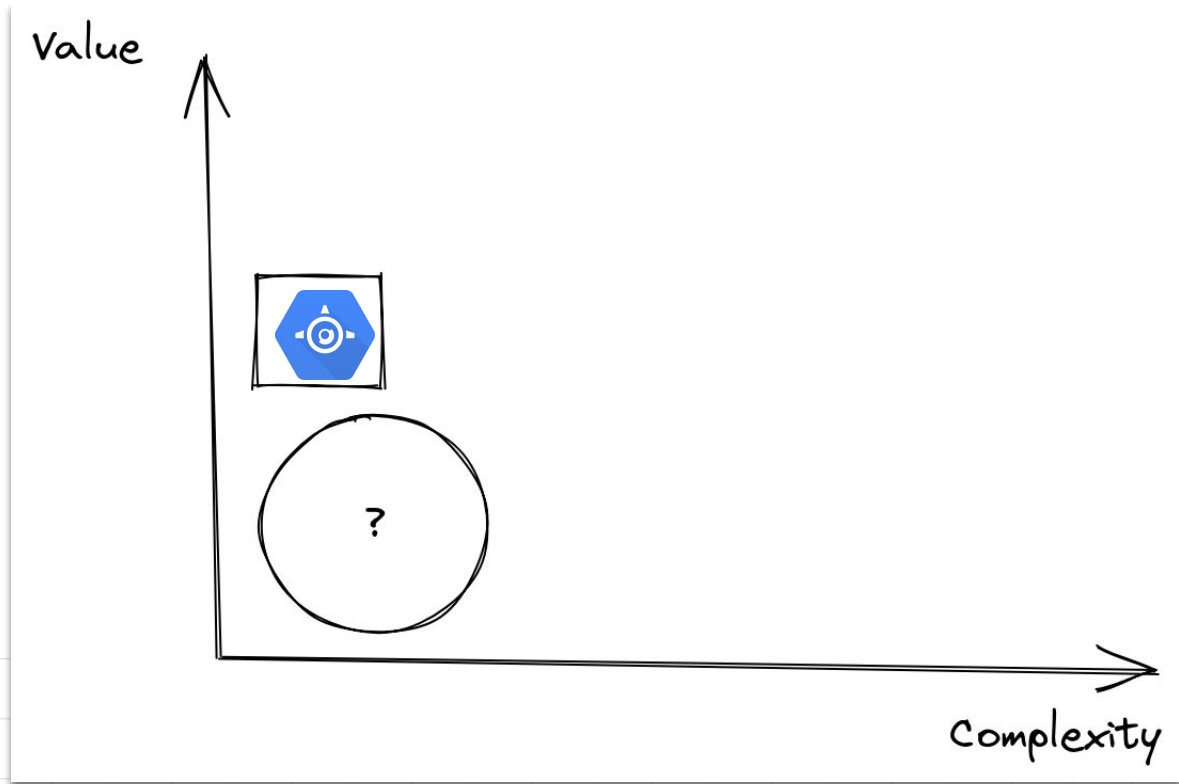
Traffic splitting between versions

Embed a stateful solution (datastore)

Automatic CDN

Scaling from 0 to N

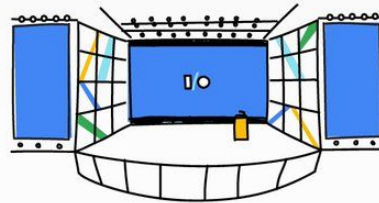
Value x Complexity



App Engine

2008

Introducing Google I/O and Google App Engine

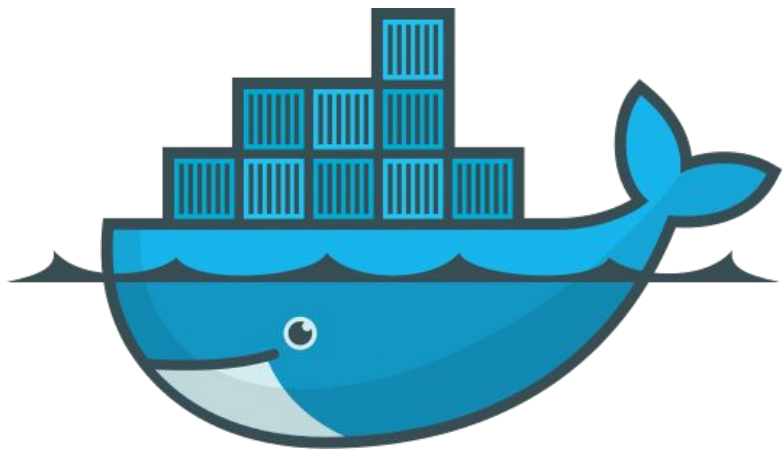


Happy 20th!



Containerization

Docker in 2016



docker

Docker

Universal package manager

Dockerfile syntax

Installation of Docker & CLI

Daemon options & linux concepts

Security

Open Source

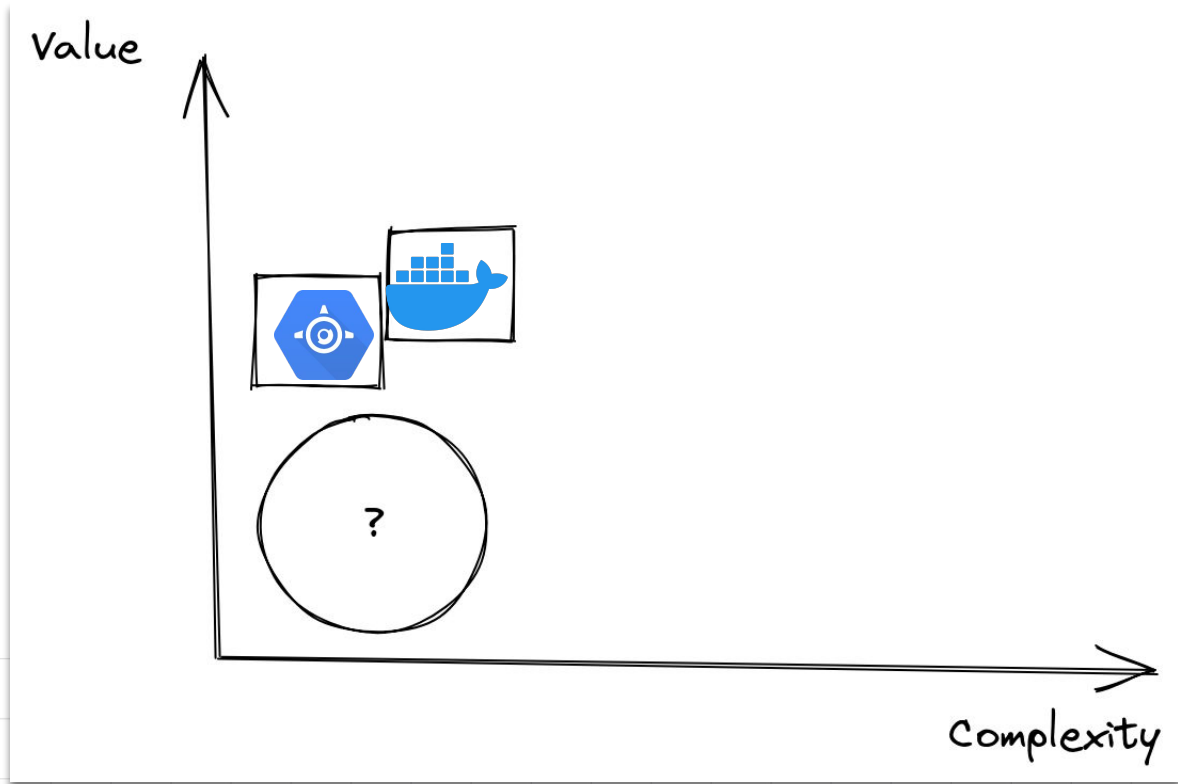
Reproducible & ephemeral concepts

Work with any runtime

Start containers quickly

Very Popular: many images & cloud
services available

Value x Complexity





#TuxTurns30

Join us in
celebrating
**30 years of
Linux**
1991 - 2021

2017: AKS & EKS

2016: Kubernetes
adoption by CNCF

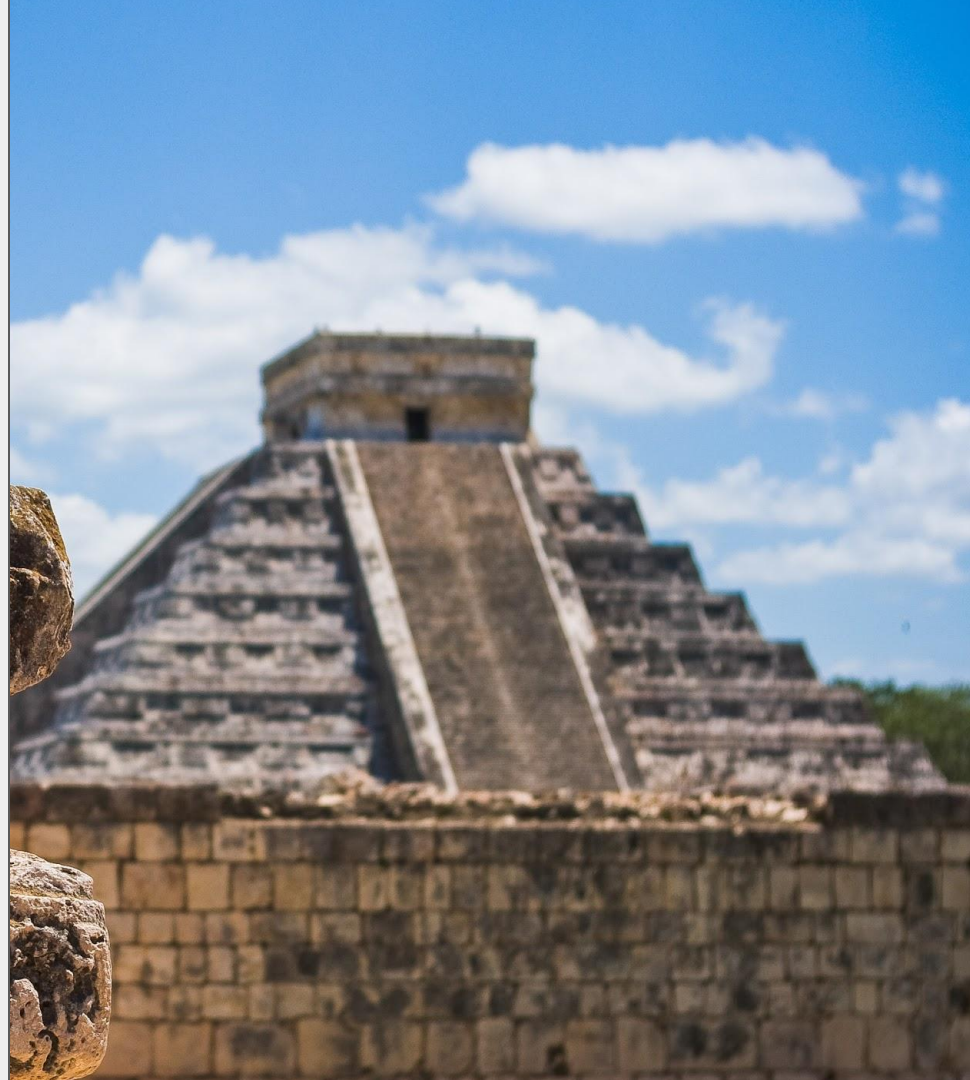
2015: GKE

2014: Kubernetes

2013: Docker

2008: LXC

2007: cgroups



Orchestration

Kubernetes in 2018



kubernetes

Kubernetes

Container orchestration at scale

New concepts: pods, services etc.

Yaml syntax

Popular: well documented

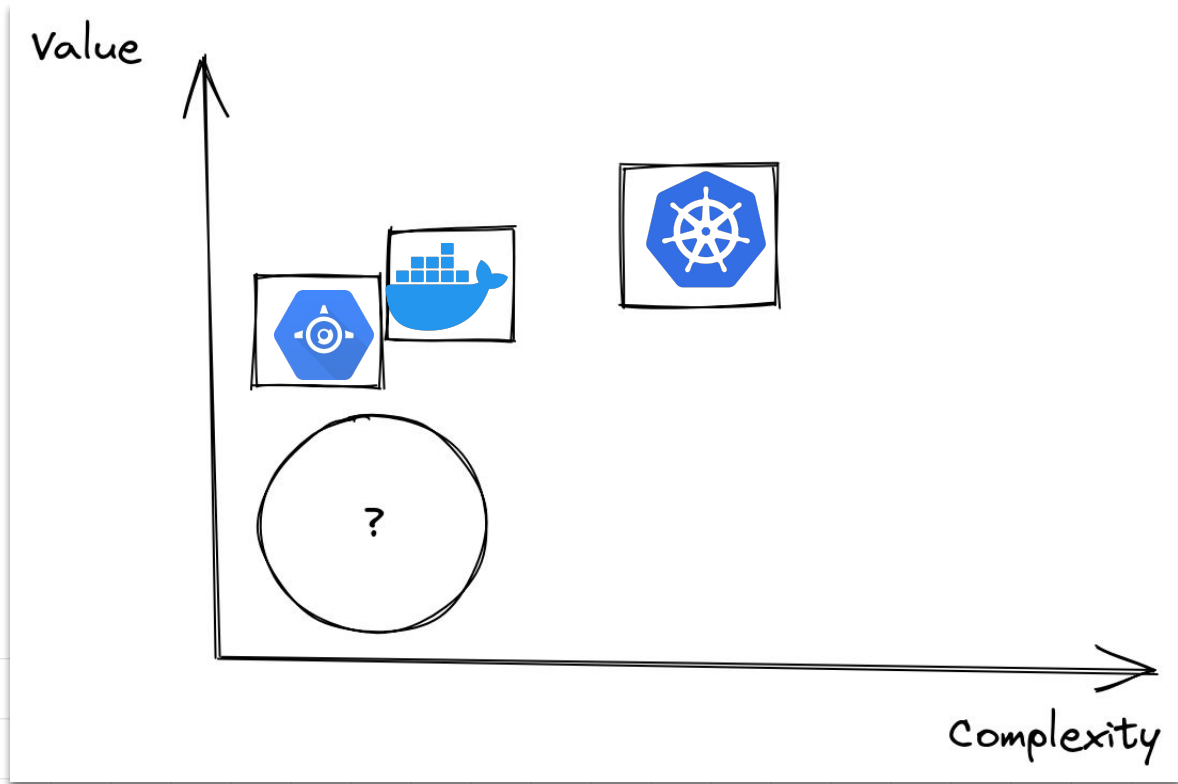
Supported by all cloud providers

Made to solve scaling issues

Pet vs Cattle

Desired State vs Actual State

Value x Complexity



Wait!

Kubernetes can do so much...

Kubernetes... again

Container orchestration at scale

Automated rollouts and rollbacks

Health checking

Load Balancing

Custom domain & traffic splitting

with Traefik

Service Mesh *with Istio*

Scale from 0 *with Knative or KEDA*

Runtime security *with Falco*

Upgrade components

Stateful objects

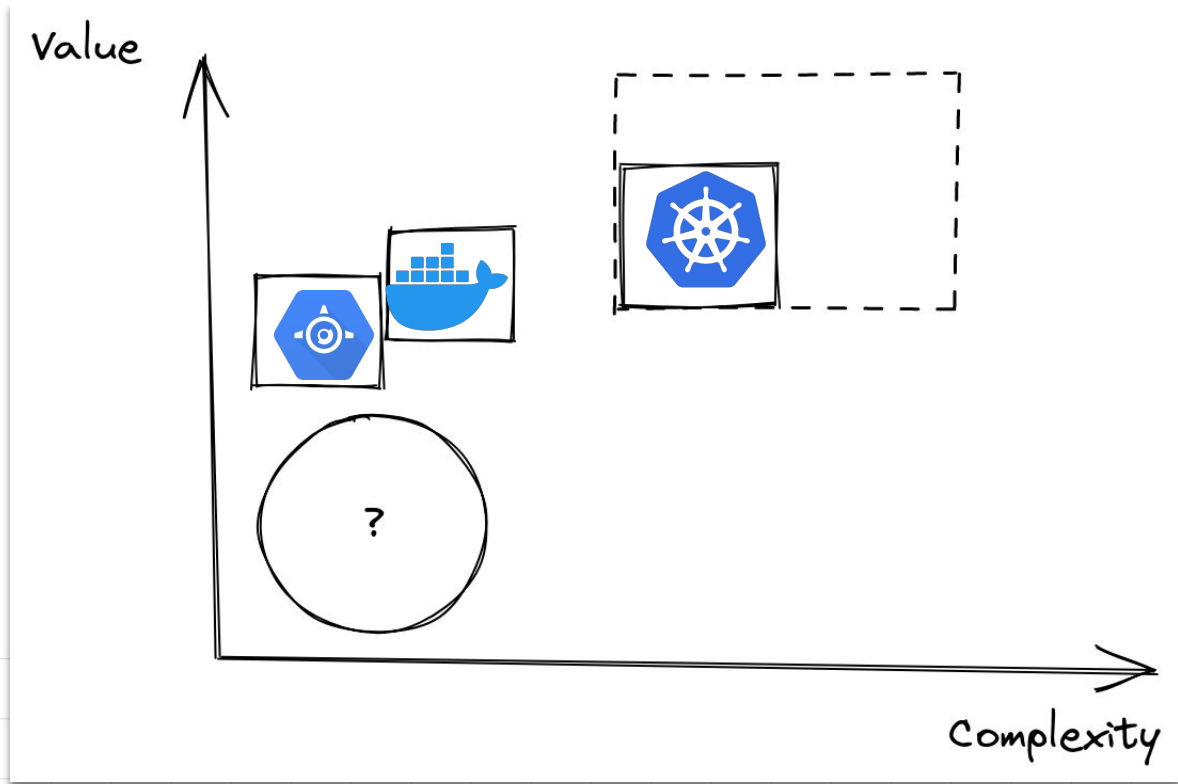
Backup

Security

IAM

Pricing

Value x Complexity



Kubernetes



Serverless



Cloud Run in 2019

Cloud Functions



App Engine



Cloud Run





Cloud Run

One of the first serverless container solution

HTTP only (gRPC)

Need IaC to deploy a complete infra

Container contract

- Listen on 0.0.0.0
- HTTP server must start < 4 min
- Request time < 60 min
- Stateless
- No background activity

Pricing Model

Easy deployment

Custom domain name

Traffic splitting between versions

Portability of containers

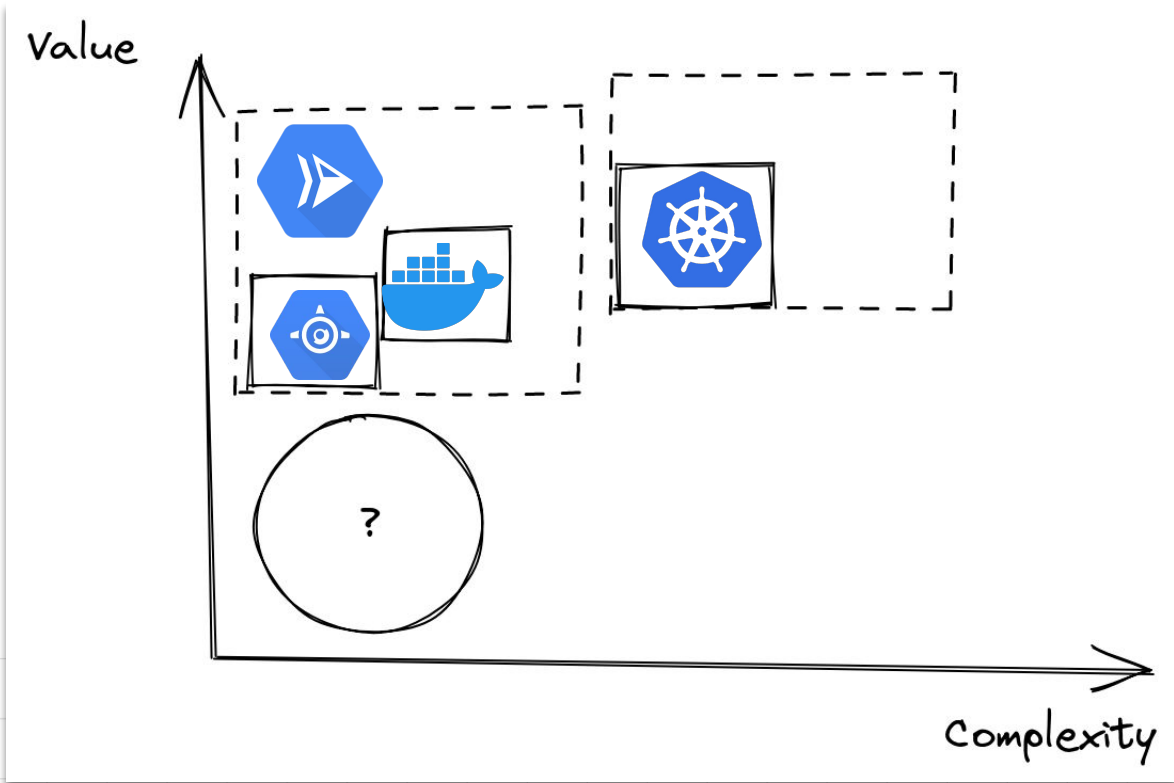
Buildpacks available

Scaling from 0 to N

No vendor lock in with Knative



Value x Complexity





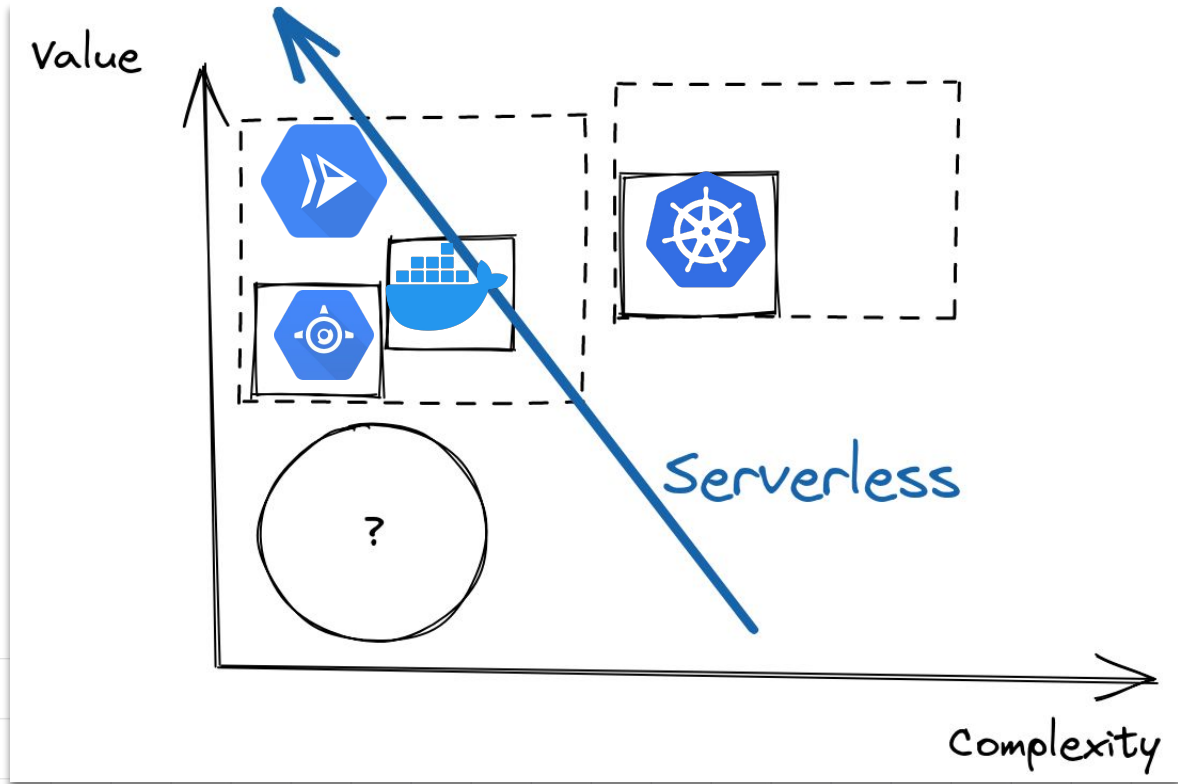
Serverless definition

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



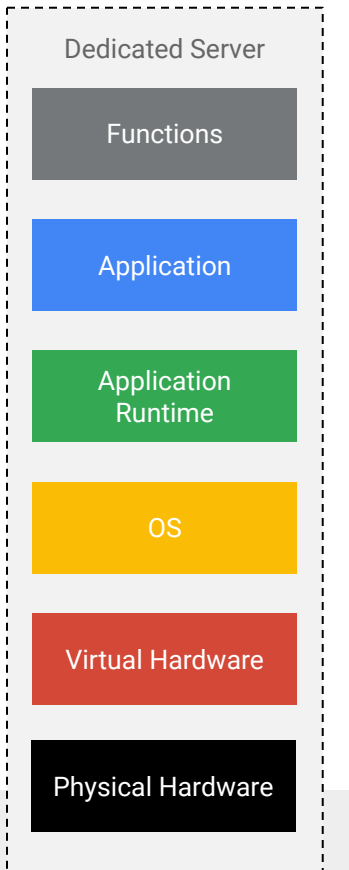


Cloud Run

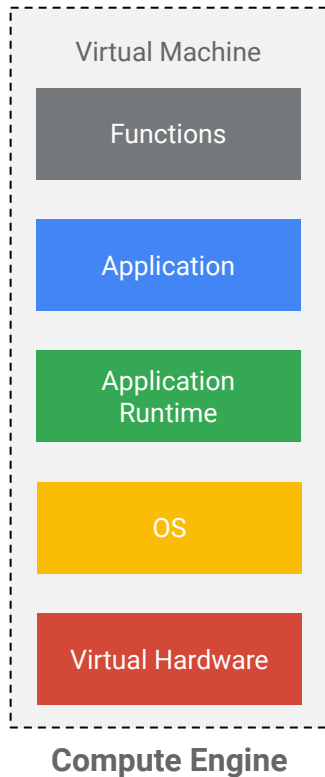




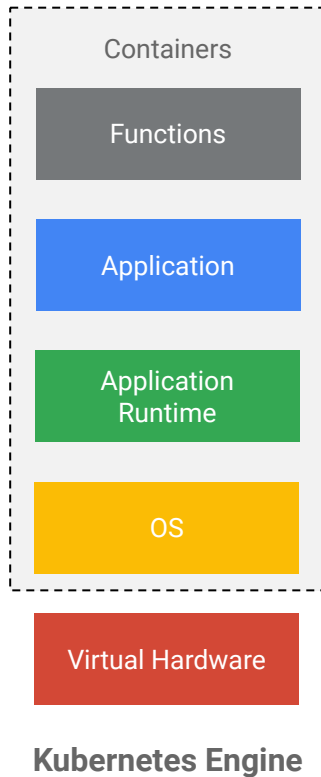
On-Premises



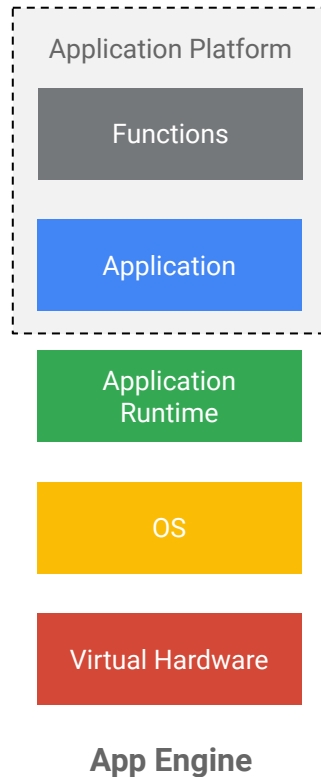
Infrastructure



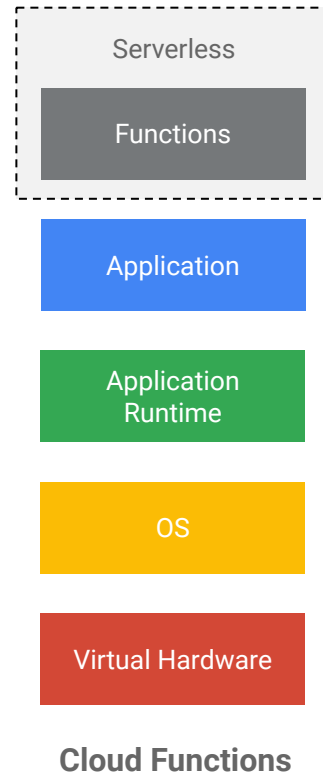
Container



Platform



Function



Cloud Run

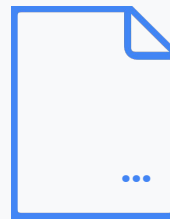
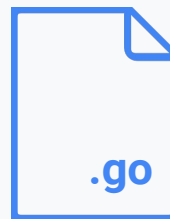
Containers

Any language

Any library

Any binary

Ecosystem of
base images

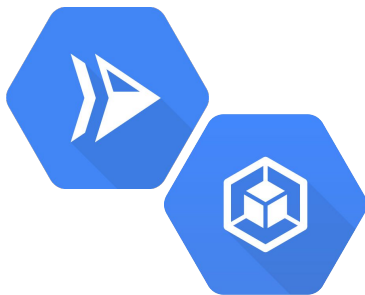


Serverless containers with Knative & Cloud Run



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)*

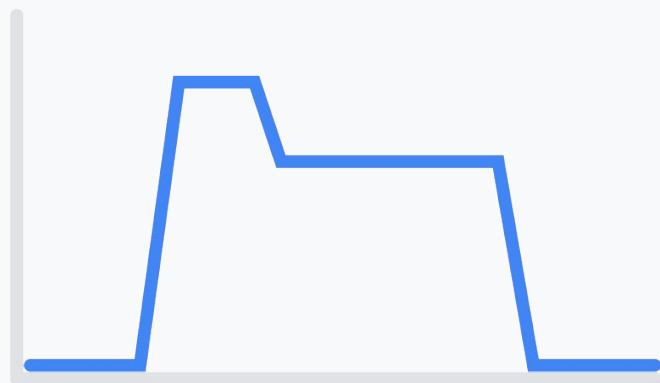
80 concurrent requests per container *(configurable 1-1000)*

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

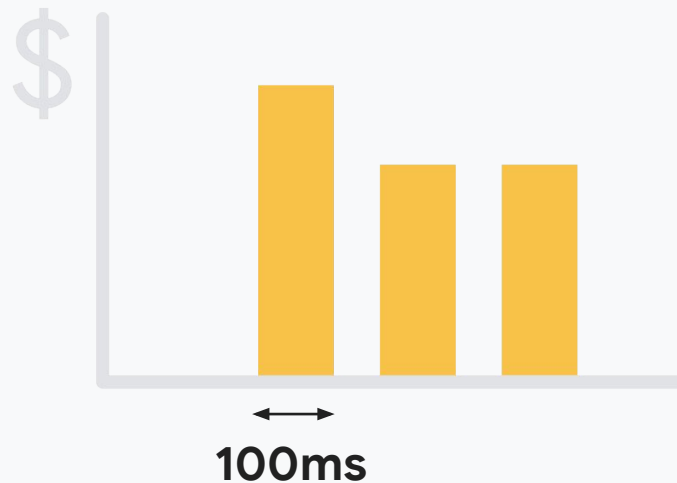
Access to a Metadata Server

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



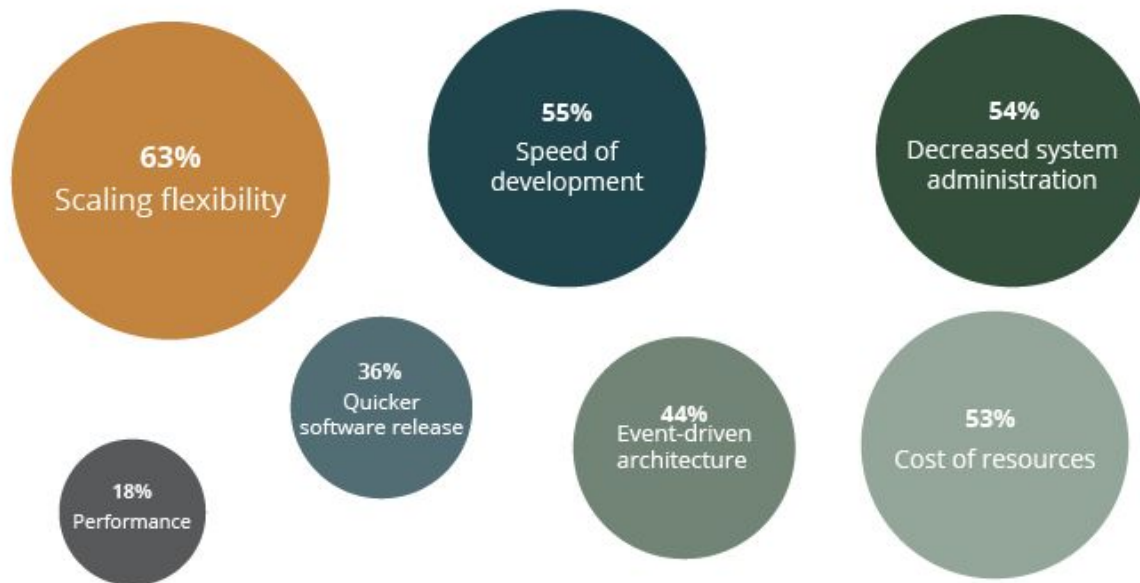


My point of view

- Cloud Run covers the majority of app use cases
- One of the TOP project Google Cloud
- Replacement of my beloved App Engine
- Interoperable (supported by Knative/eventing)
- Superior Dev Experience (local emulator & buildpacks)



Serverless trends





To sum up

- 👍 Kubernetes and Serverless can support each other
- 😲 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
- 💪 The right tool to do the right thing
- 🤔 Find your way: more a Dev (Serverless World) or an Admin (K8S World)



Thank You!



Julien Landuré
 @jlandure
jlandure.dev

