



Evolution d'une 'Plate-forme en tant que service' Vu de l'intérieur

Ludovic Champenois
Google Cloud, App Engine Java TL
Greygler

Google Cloud



@ludoch



Plate-forme en tant que service, c'est quoi?

“Plate-forme en tant que service, PaaS, de l'anglais platform as a service, est l'un des types de cloud computing, principalement destiné aux entreprises, où :

l'entreprise cliente maintient les applications proprement dites ;
le fournisseur cloud maintient la plate-forme d'exécution de ces applications ...

Une première vague de services de ce type a vu son apparition vers 2006-2008 avec Heroku, Engine Yard ou **Google App Engine**, une seconde vague a vu son apparition avec la démocratisation des Conteneurs Linux autour de 2014 par le projet open-source Docker créé par l'entrepreneur franco-américain Solomon Hykes.

Cette deuxième vague tend à se confondre avec un autre mouvement contemporain le **Serverless Computing** (l'informatique sans serveur) qui propose des promesses similaires (déploiement rapide de code sans la nécessité de configurer l'infrastructure sous-jacente) ...”



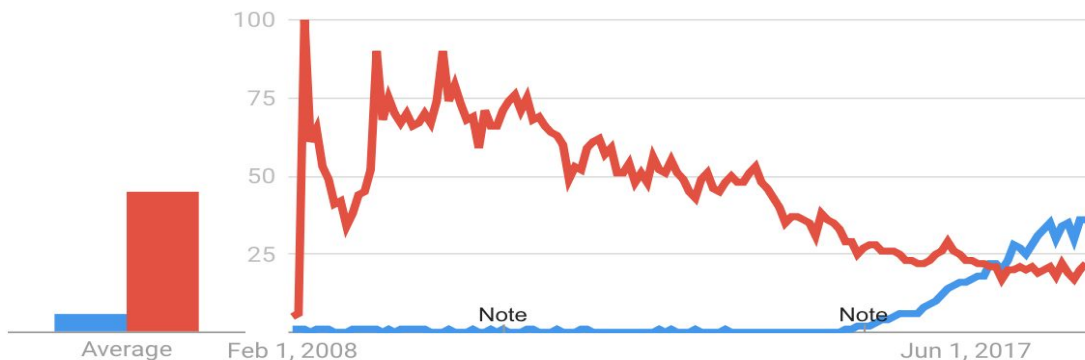
https://fr.wikipedia.org/wiki/Plate-forme_en_tant_que_service

Serverless/'Sans Serveur', c'est quoi?

Interest over time

Google Trends

● serverless ● Google App Engine



Serverless Runtimes^{[[edit](#)]}

Most, but not all, serverless vendors offer compute runtimes, also known as [function as a service \(FaaS\)](#) platforms, which execute application logic but do not store data. The first "pay as you go" code execution platform was Zimki, released in 2006, but it was not commercially successful.^[3] In **2008**, Google released [Google App Engine](#), which featured metered billing for applications that used a custom Python framework, but could not execute arbitrary code.^[4] PiCloud, released in 2010, offered FaaS support for Python.^[5]

Serverless, c'est quoi?

Modèle Opérationnel



Pas de Management de l'infra



Sécurité géré



coût à l'usage

Modèle de Programmation



Basé sur des Services



Événementiel



Ouvert

Serverless: les origines

guido@ [Novembre 2008](#):

“Unlike other cloud offerings, App Engine does not offer you a virtual machine, but a scalable **container** in which your application runs...”

[Camp Fire video](#)

Présentation Originale:

web.stanford.edu/class/ee380/Abstracts/081105-slides.pdf

Novembre 2008

Et oui, 2 11 ans déjà...



Google IO 2009

What is Google App Engine?

- A cloud-computing platform
- Run your web apps on Google's infrastructure
- We provide the container and services (PaaS)
 - Hardware, connectivity
 - Operating system
 - JVM
 - Servlet container
 - Software services

Google I/O 2009 - App Engine: Now Serving Java

19,705 views

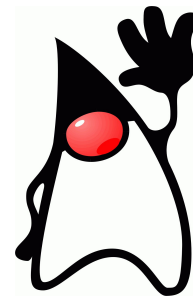
Google Developers ©
Published on Jun 2, 2009

SUBSCRIBE 1.7M

Cloud-Computing

Infrastructure

Container



Source: www.youtube.com/watch?v=ofn8QYEVyhA

Google IO 2009

Java Support

- Servlets
- Software services
- Sandboxing
- DevAppServer
- Deployment
- Tooling



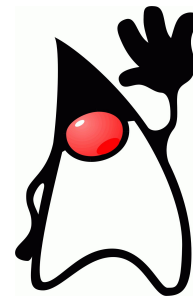
Google I/O 2009 - App Engine: Now Serving Java

19,705 views

Google Developers
Published on Jun 2, 2009

SUBSCRIBE 1.7M

Sandboxing...



Source: www.youtube.com/watch?v=ofn8QYEVyhA

Google IO 2009

Sandboxing

- What do we do?
 - Restrict JVM permissions
 - WhiteList classes
- Why is it necessary?
 - Clustering - JVMs come and go
 - Protect applications from one another
 - Quality of service

Google IO

16:33 / 55:00

Google I/O 2009 - App Engine: Now Serving Java

19,705 views

38 6 SHARE SAVE ...

Google Developers
Published on Jun 2, 2009

SUBSCRIBE 1.7M



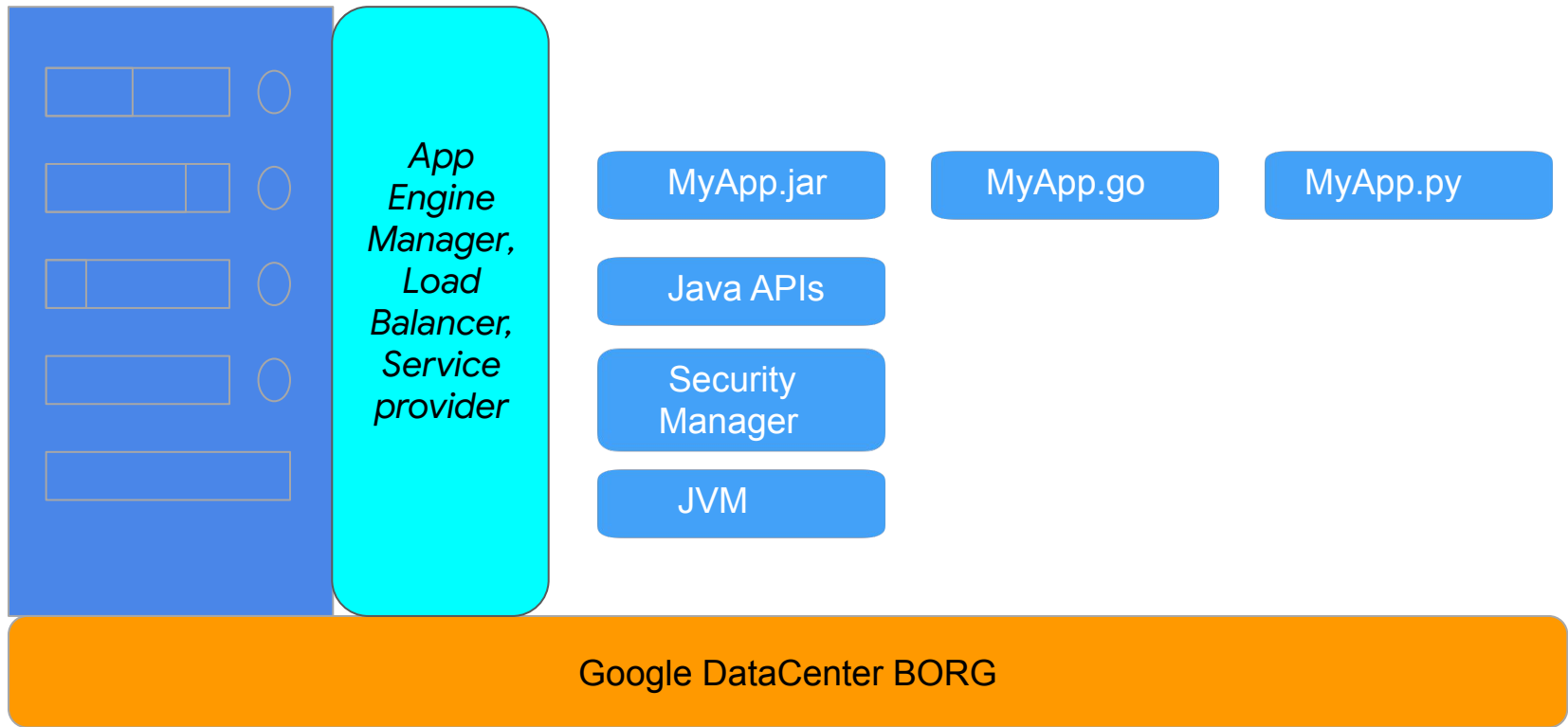
Sandboxing:

- Restrict JVM
- Whitelist classes
- ...

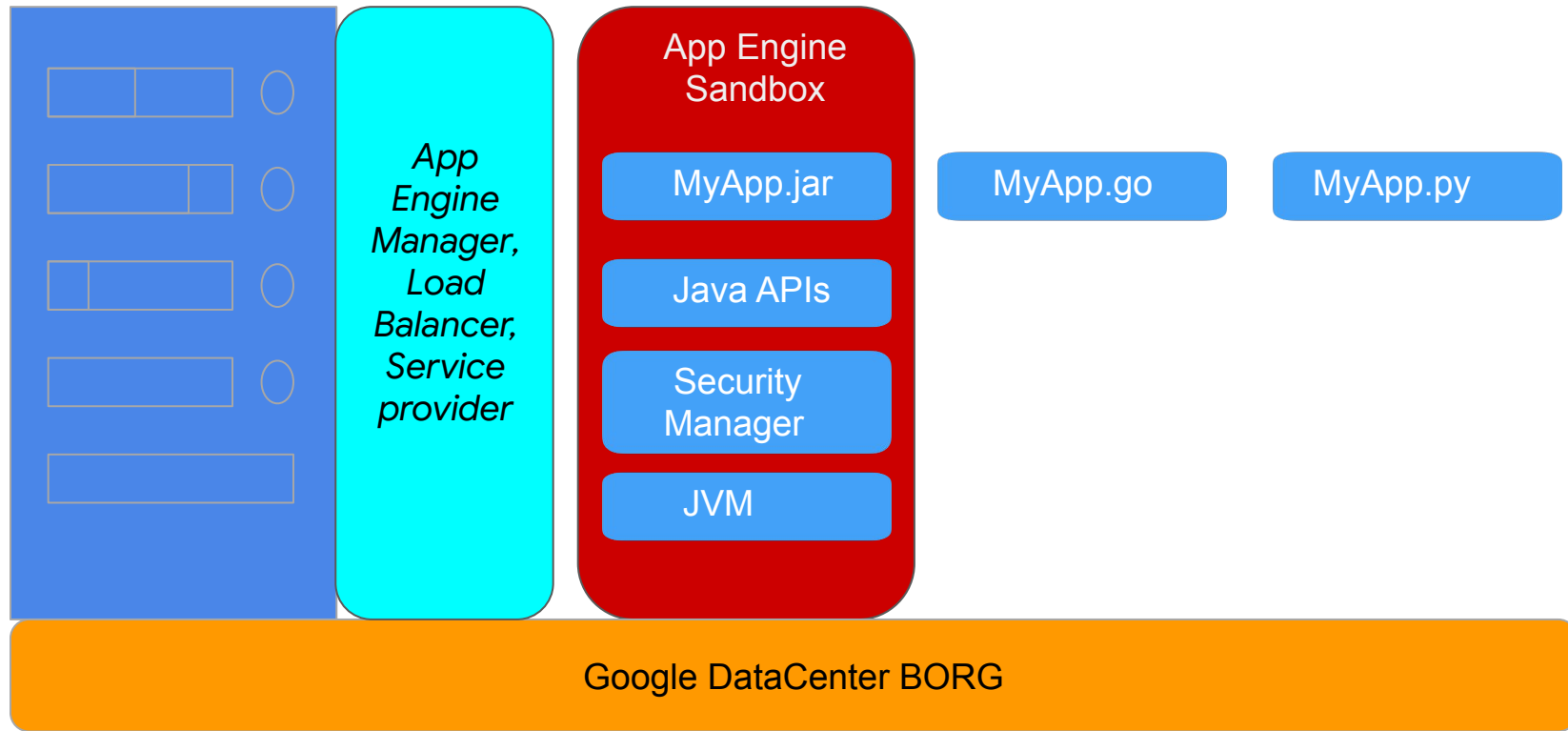


Source: www.youtube.com/watch?v=ofn8QYEVyhA

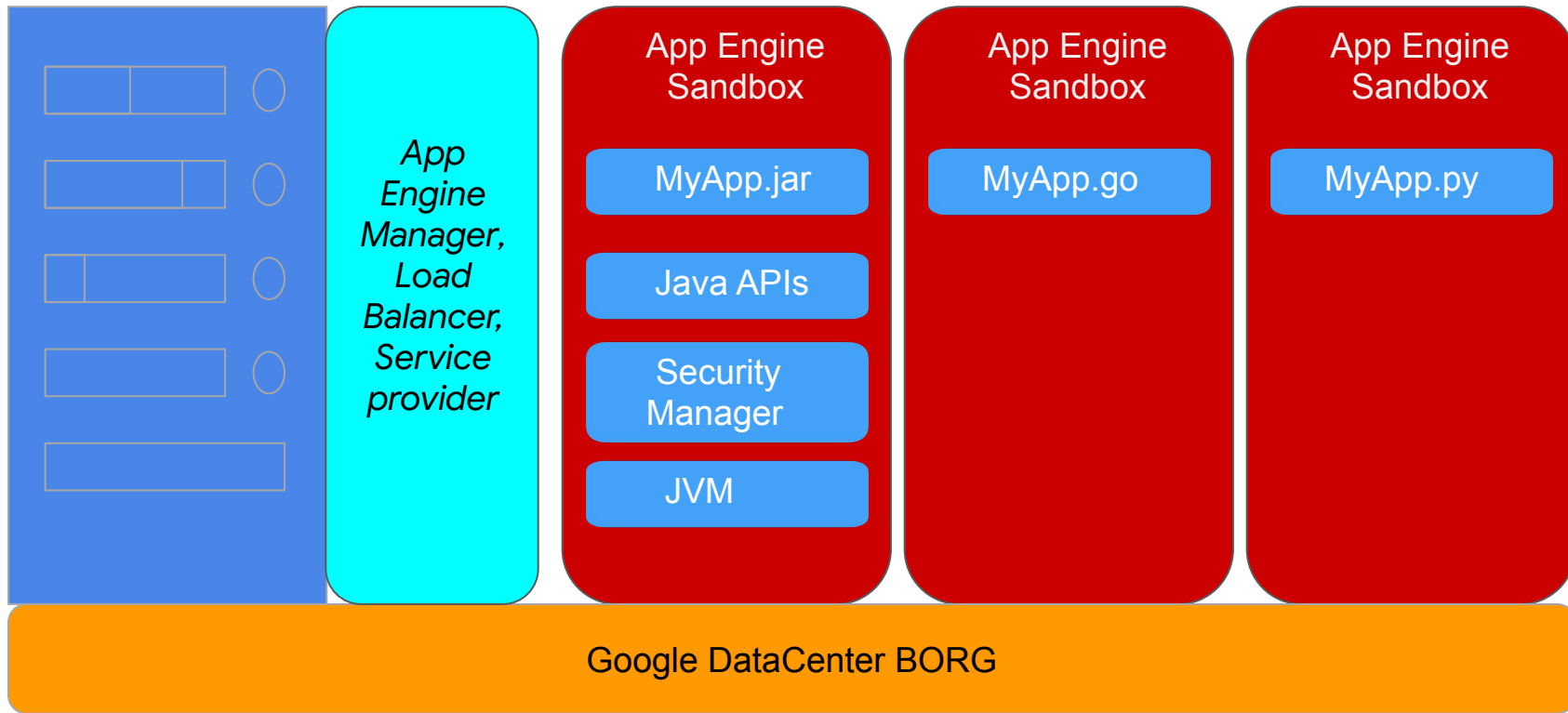
Une Sandbox: pourquoi?



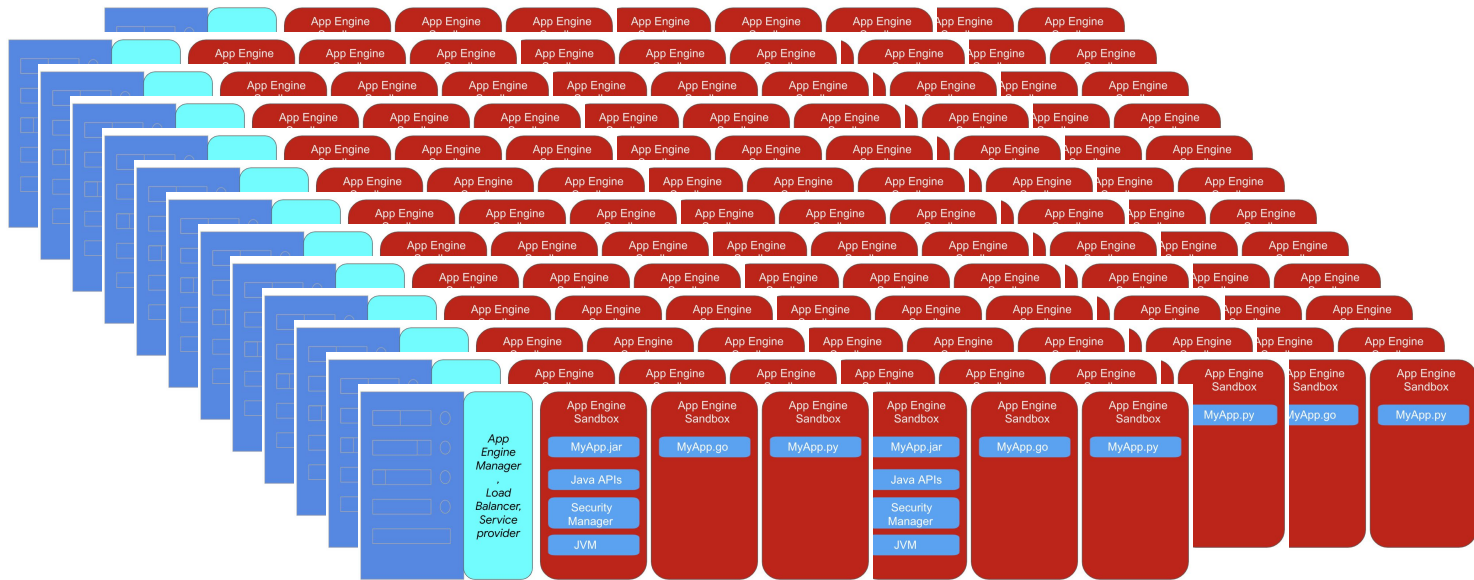
Une Sandbox: pourquoi?



Une Sandbox: pourquoi?



Une Sandbox: pourquoi?

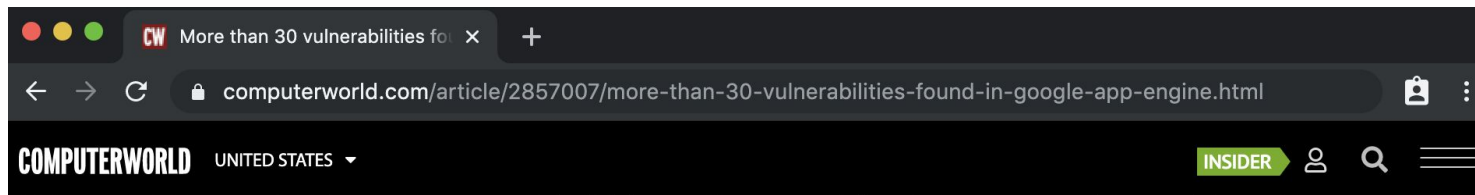


Google DataCenter BORG

PaaS: Ma Définition:



'Sandbox' Restrictive, pourquoi?



More than 30 vulnerabilities found in Google App Engine

Researchers escaped the Java sandbox on the cloud platform and executed code on the underlying system



By Lucian Constantin

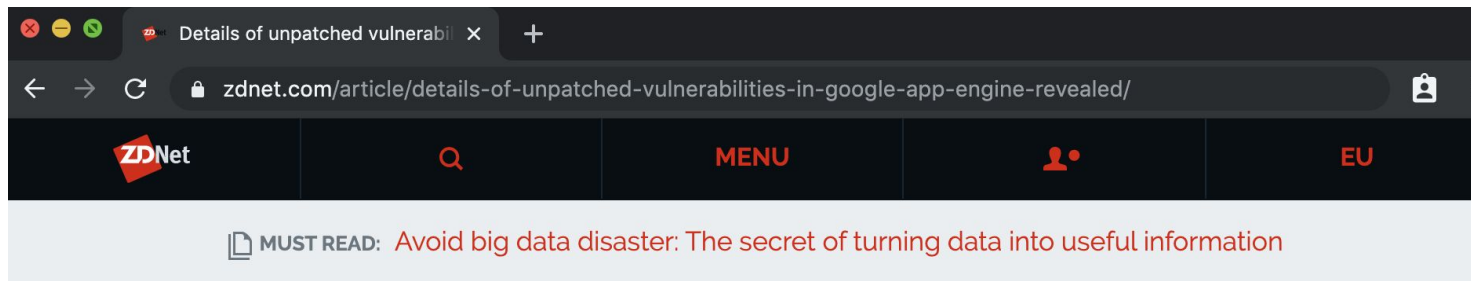
CSO Senior Writer, IDG News Service | DEC 9, 2014 11:31 AM PST

www.computerworld.com/article/2857007/more-than-30-vulnerabilities-found-in-google-app-engine.html

Dec 2014



Sandbox Réstrictive, pourquoi?



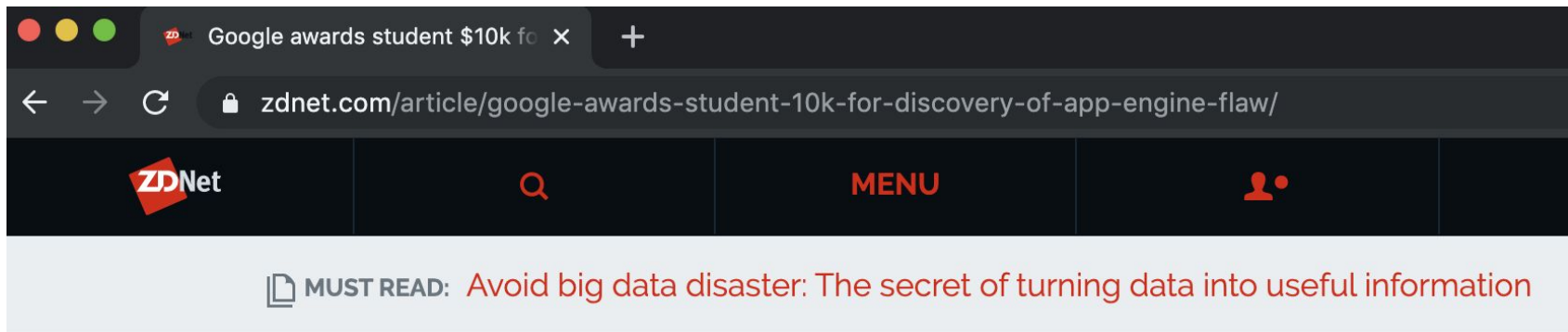
Details of unpatched vulnerabilities in Google App Engine revealed

Google is known for playing hardball when it comes to firms fixing security problems -- and now the company itself is being held under the same standard. [UPDATED]

www.zdnnet.com/article/details-of-unpatched-vulnerabilities-in-google-app-engine-revealed/

May 2015

Sandbox Restrictive, pourquoi?



Google awards student \$10k for discovery of App Engine data leak flaw

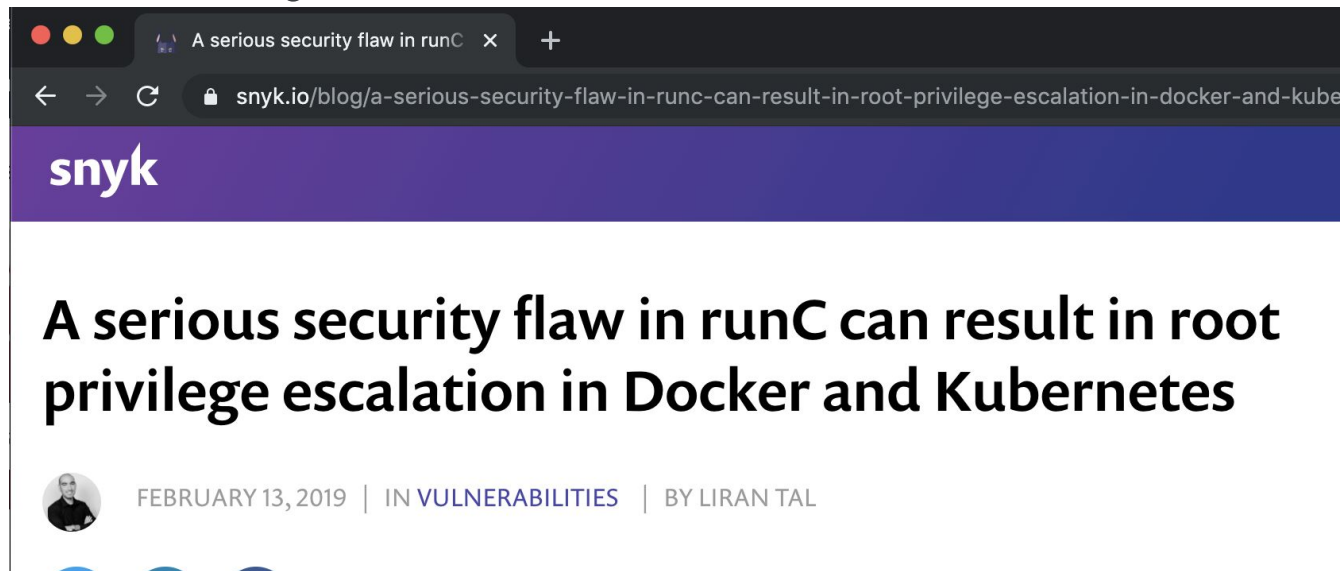
An Uruguayan student found a bug which could have allowed the leak of sensitive data.

www.zdnet.com/article/google-awards-student-10k-for-discovery-of-app-engine-flaw

August 2017



Mais moi, je suis sur Docker, donc ça va, hein?



<https://snyk.io/blog/a-serious-security-flaw-in-runc-can-result-in-root-privilege-escalation-in-docker-and-kubernetes/>

August 2017

gVisor à la rescousse!

OpenJDK 8 est arrivé en 2014...La GAE sandbox originelle ne tient plus..

App Engine le supporte, **3 ans après**, grâce à:



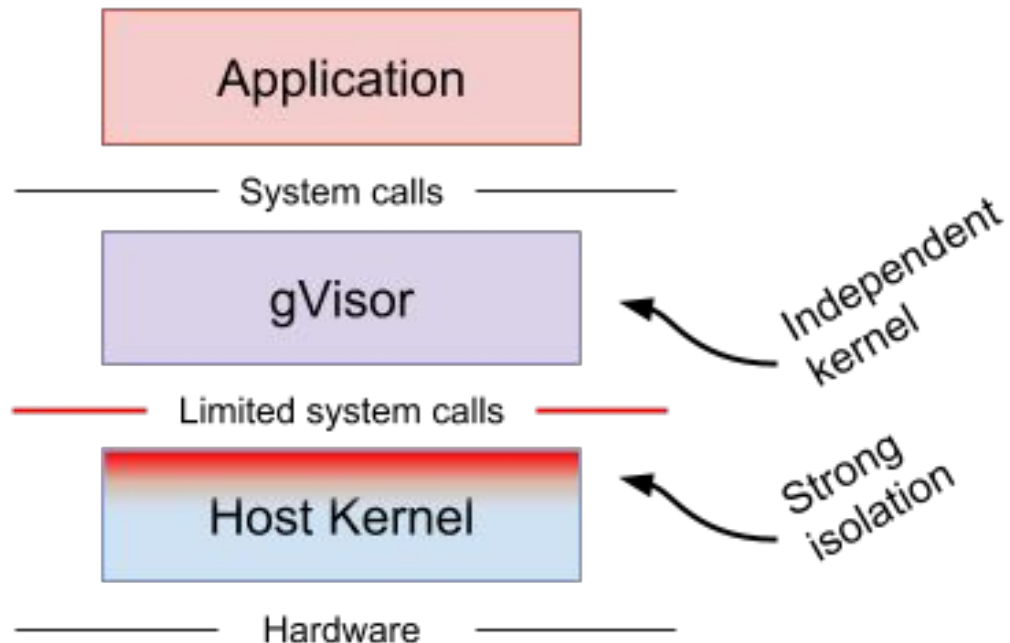
gVisor, Késako?

gVisor est un ‘user-space kernel’, écrit en Go, qui implémente une surface importante d’un système **Linux**.

gVisor intercepte les **appels système** de l’application et agit comme un noyau ‘guest’.

Au début, hautement  chez Google,
Maintenant Open Source... gvisor.dev/

gVisor, Késako?



gVisor, Késako?

- Une sandbox basé sur la virtualisation.
- Isolation forte et sécurité garantie (source ouvert).
- Ce n'est **pas** une machine virtuelle.
- Implémente la plupart des Linux syscalls (250++).
- Léger et rapide.
- Plus besoin de changer la JVM/Libraries.

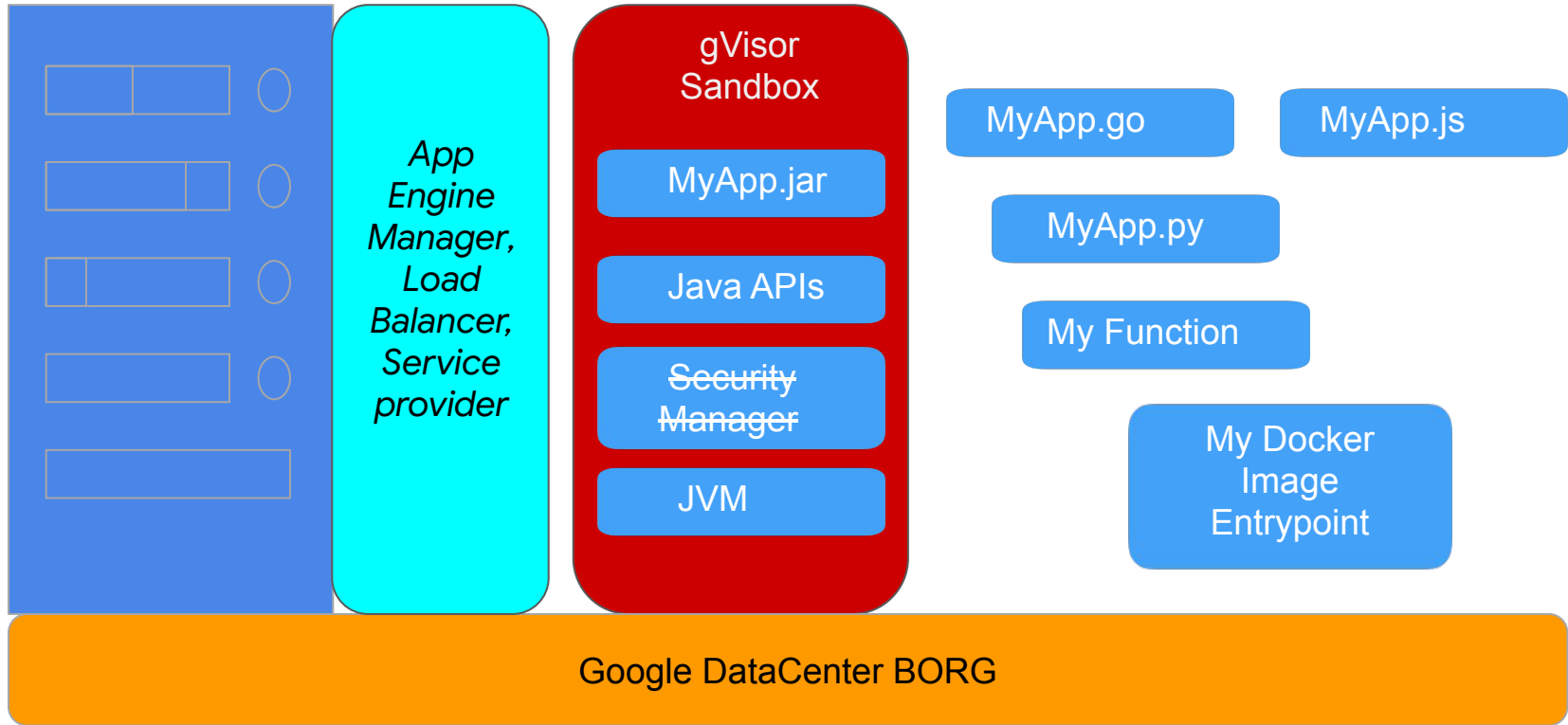
gVisor et les conteneurs

- OCI-compatible
- S'intègre avec Docker, containerd, Kubernetes!

gVisor: App Engine Java8

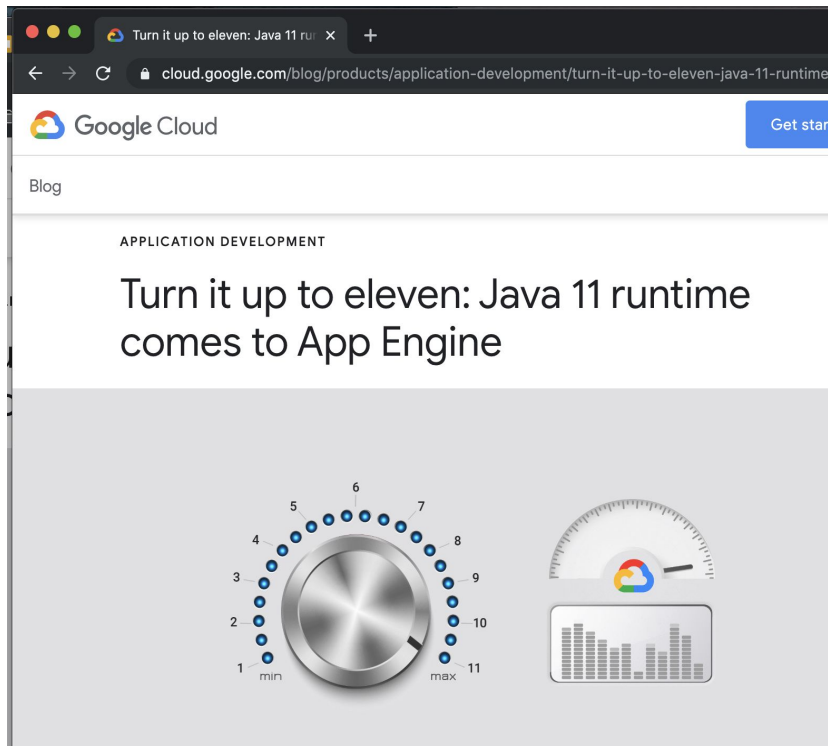
- GAE Java8 est sorti en Septembre 2017
- Compatibilité très strict pour des Millions de Web Apps
- **Sans** les limitations initiales
- Supporte les APIs Standard App Engine et les APIs Cloud
- Des **Millions** de `Queries Per Second` (QPS)
- OpenJDK 8 et Jetty 9 (Servlet 3.1 Web Apps)
- **Migration** automatique et transparente de Millions d'apps Java 7

gVisor: ~~JVM security Manager~~



11/2019: App Engine Java11

- GAE Java11 **GA** à Devovx 2019
 - Ouvert, sans contraintes, 100% managé
 - Vous fournissez un Web Serveur
-
- `$ gcloud app deploy myjar.jar`
 - `$ gcloud app deploy pom.xml (soon)`



Demo

GAE Java11



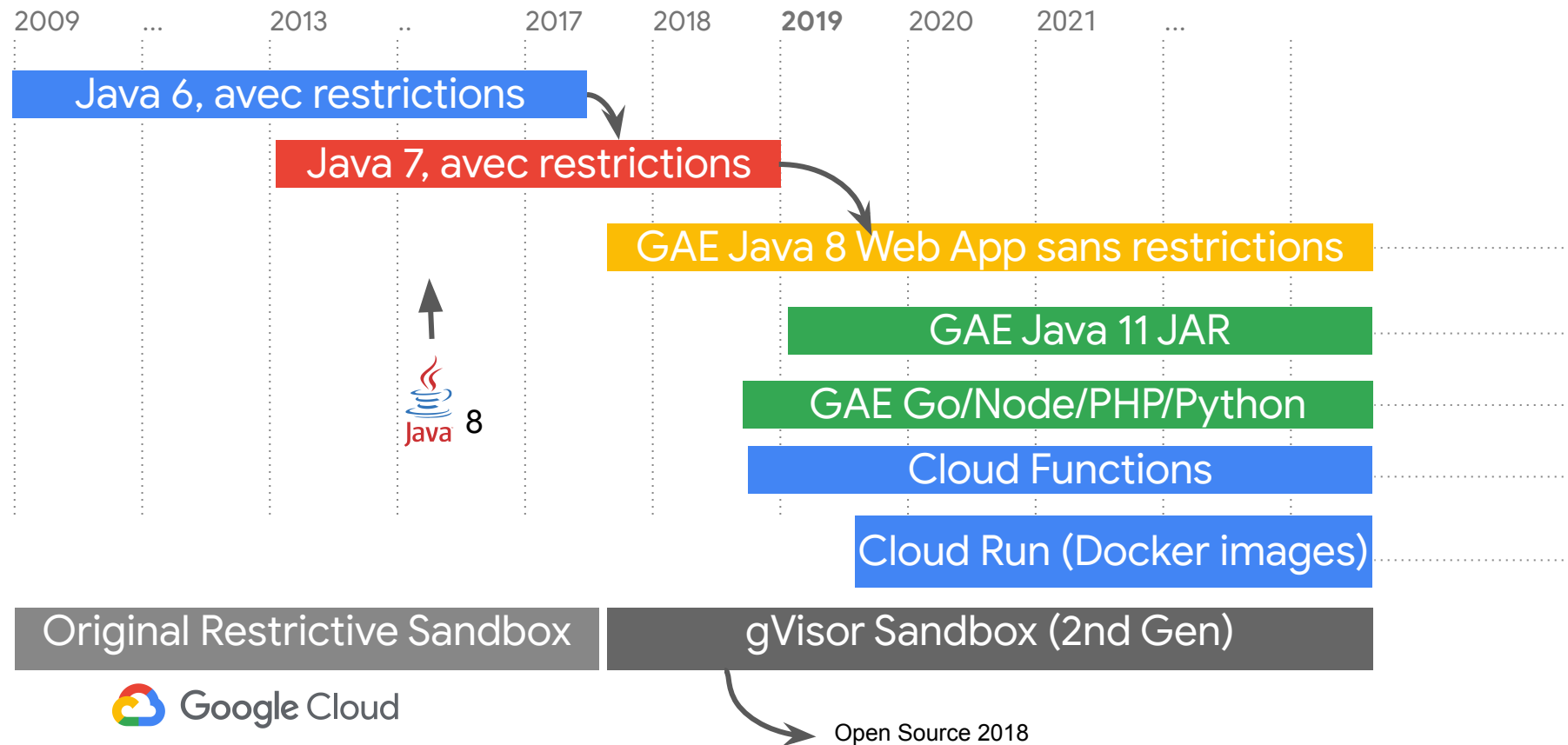
App Engine Java11

Runs	Des JARs avec serveur sur port 8080 ou \$PORT Aussi des GraalVM applications natives
Framework	Tous! Spring Boot, Micronaut, Quarkus, Vert.x, ...
JVM Language	Tous! Kotlin, Groovy, Scala, ...
Cloud API	Toutes (Cloud)! Datastore, Tasks, Firestore, Pub/Sub,...
Databases	Toutes!

bit.ly/appengine-java11-blog



Evolution du ~~PaaS~~ Serverless Google



Annonçant **Cloud Run**

Déploiement d'image de conteneurs



Cloud Run



**Du Container à la
production en quelques
secondes**



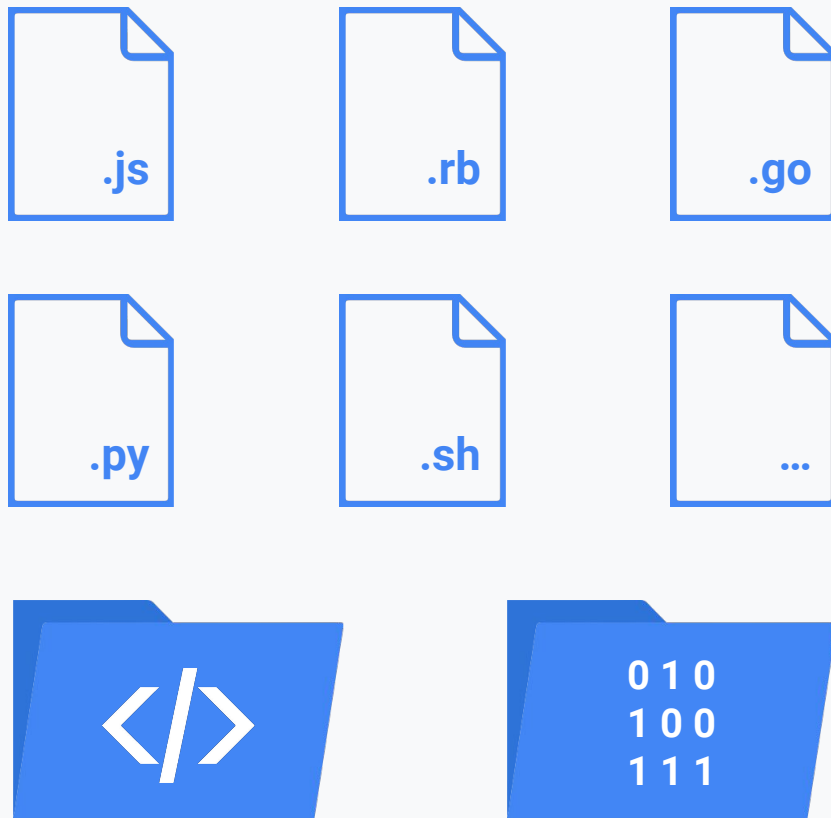
Serverless Native



**expérience simple,
ou vous voulez**

Conteneurs


- Tout langage
- Toute librairie
- Tout binaire
- Eco-Système
d'images de bases
Docker



App Engine

ou

Cloud Run?

JAR(s) ou pom.xml, ou Application code (Native image GraalVM...)	Container Image
Image de base entièrement managé	Toute Docker image (votre responsabilité)
OpenJDK 11.x (x toujours mis à jour)	Votre propre OpenJDK ou binaire (votre responsabilité)
Exécution automatique du JAR, aussi custom	Vous fournissez un Docker entrypoint
Traffic Spitting, URL Mapping, VPC, Cloud Debugger,...	Cloud Run pour Anthos, VPC, unauthenticated invocations or not,...
app.yaml fichier de configuration(optionnel)	Configuration via CLI or Console
<code>gcloud app deploy app.yaml myapp.jar pom.xml</code> (pom.xml -> GAE standard Buildpacks)  Google Cloud	<code>gcloud run deploy --image=gcr.io/monproject/monimage:montag</code>

	App Engine	Cloud Run
Ingress / HTTPs	Oui	Oui
Custom Domain	Oui	Oui
URL Mapping	Oui (dispatcher)	Non (service par sub-domain)
Traffic Splitting	Oui	
Min / Max Instances	Oui	
Autoscaling	Oui (CPU, Request)	Request based

Exemple de Cloud Run Dockerfile

```
FROM maven:3.5-jdk-8-alpine as builder
# Copy local code to the container image.
WORKDIR /app
ADD pom.xml .
COPY src ./src
RUN mvn package -DskipTests
FROM adoptopenjdk/openjdk8:jdk8u202-b08-alpine-slim
COPY --from=builder /app/target/lib /lib
COPY --from=builder /app/target/hi-1-runner.jar /hi.jar
CMD ["java","-jar","/hi.jar"]
```

Demo

Cloud Run



■ Expérience simple, où vous voulez



Cloud Run

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



Cloud Run avec Anthos

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



Knative Partout

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



Merci!

- A lire: bit.ly/appengine-java11-blog



App Engine

Google Cloud



@ludoch