Advanced Big Data:
Cloud Services & Serverless

Dan Zaratsian

AI/ML Solutions Architect, Gaming @ Google

d.zaratsian@gmail.com

https://github.com/zaratsian

TOPICS

- Session 1: Course Intro, Trends, and Approach to AI/ML
- Session 2: SQL and NoSQL
- Session 3: Distributed ML with Spark and Tensorflow
- Session 4: Cloud Generative AI Services and Architectures
- Session 5: Cloud Machine Learning Services
- Session 6: Serverless ML, Architectures, and Deploying ML

What is cloud computing?

IaaS

Infrastructure-as-a-Service

PaaS

Platform-as-a-Service

SaaS

Software-as-a-Service

Google Cloud Platform

Google Cloud

Why Serverless?

Operational Model

Invisible Infrastructure

Fully managed security

Pay only for usage

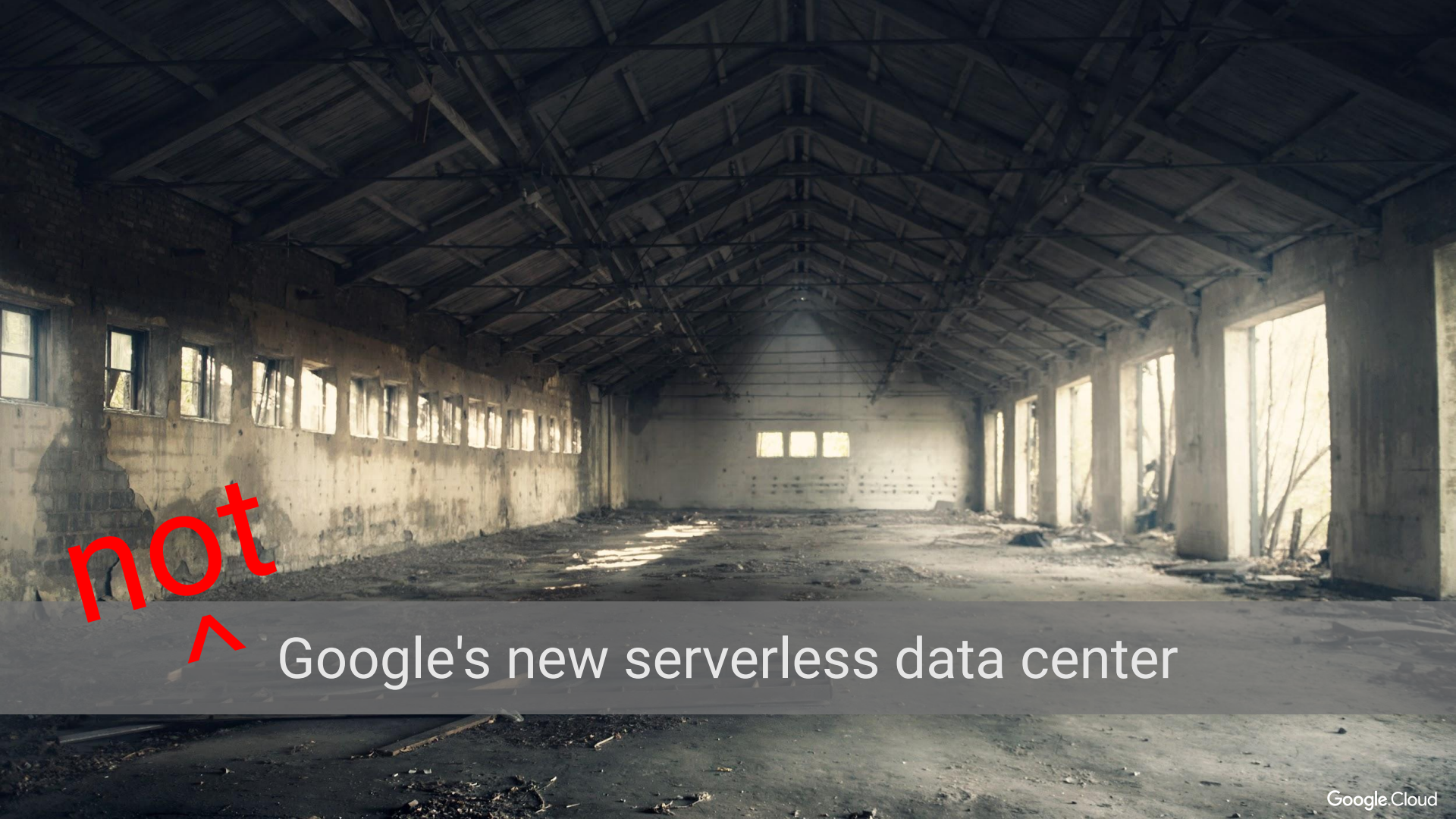
Programming Model

Microservices Event-Driven

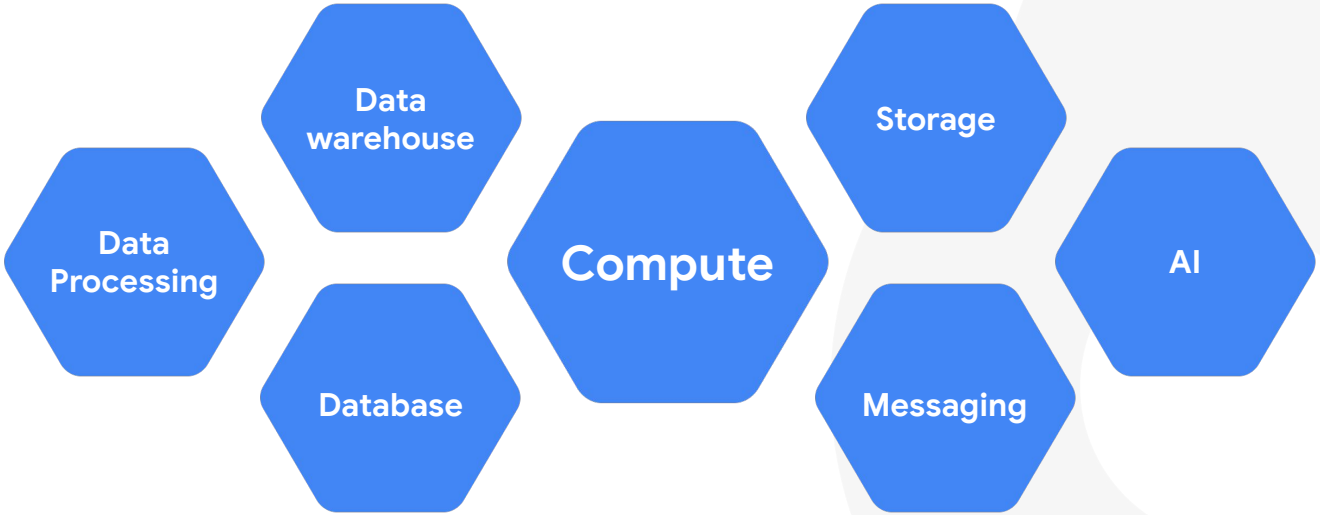
Elastic Scaling

Open

Focus on your Application - Not the Infrastructure


not
Google's new serverless data center

What is serverless…

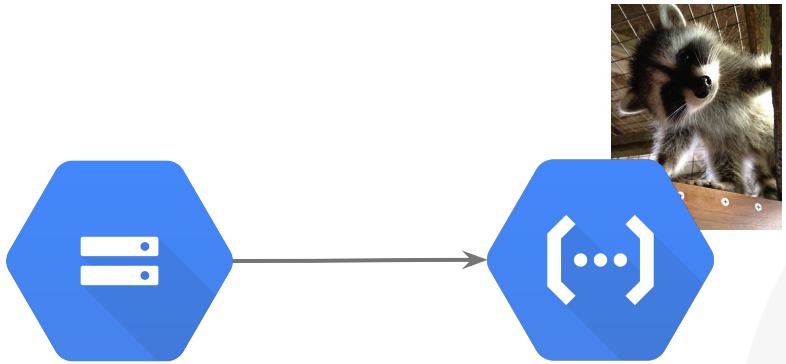


Security ● Networking ● OS ● Virtualization ● Middleware ● Runtime

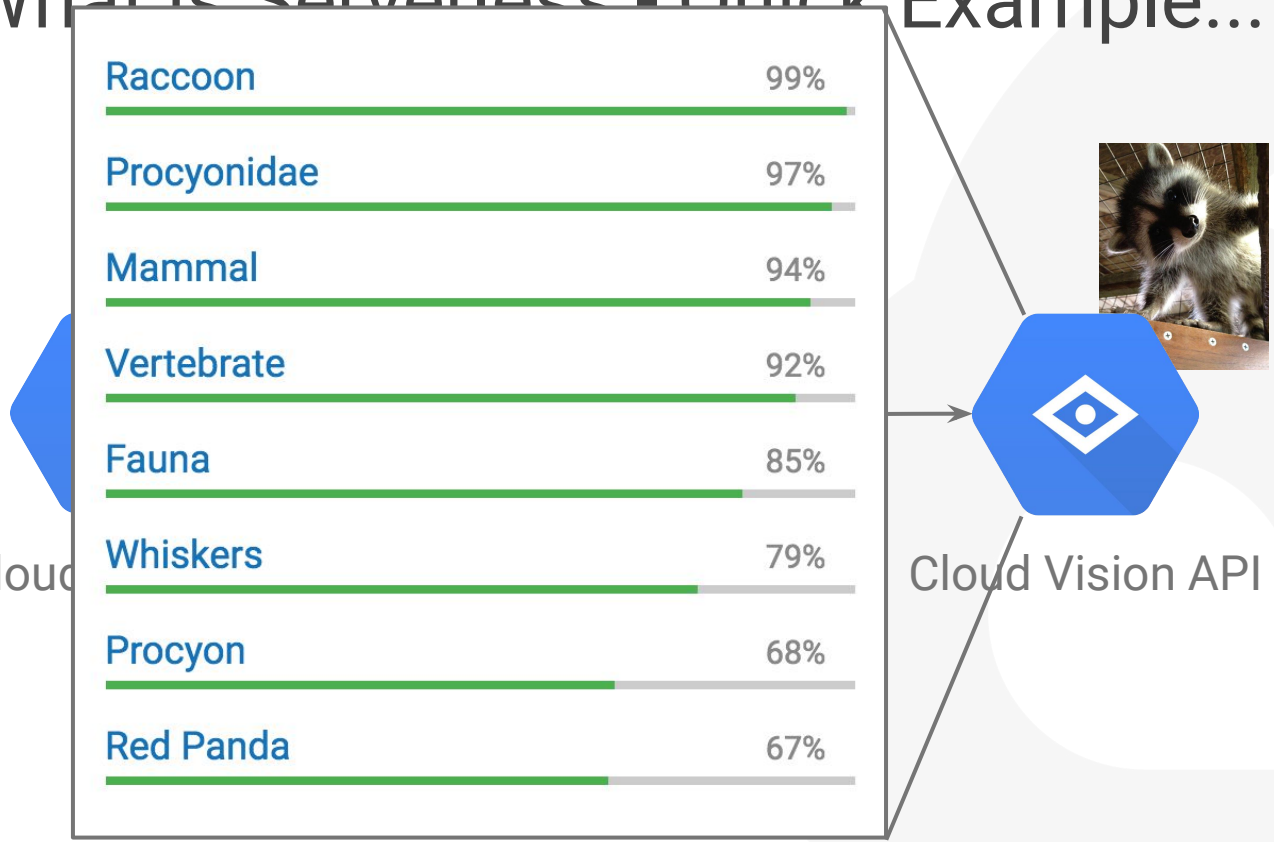
What is Serverless - Quick Example...

Cloud Storage

What is Serverless - Quick Example...

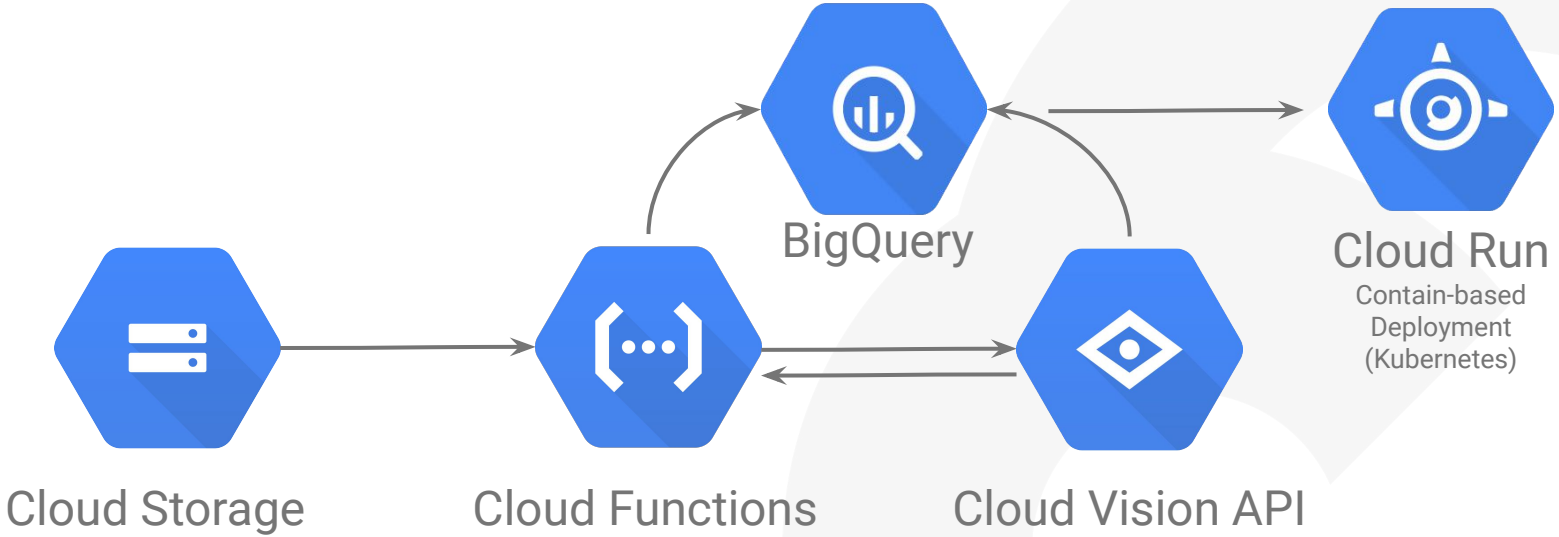


What is Serverless - Quick Example...



What is Serverless - Quick Example...

What is Serverless - Quick Example...



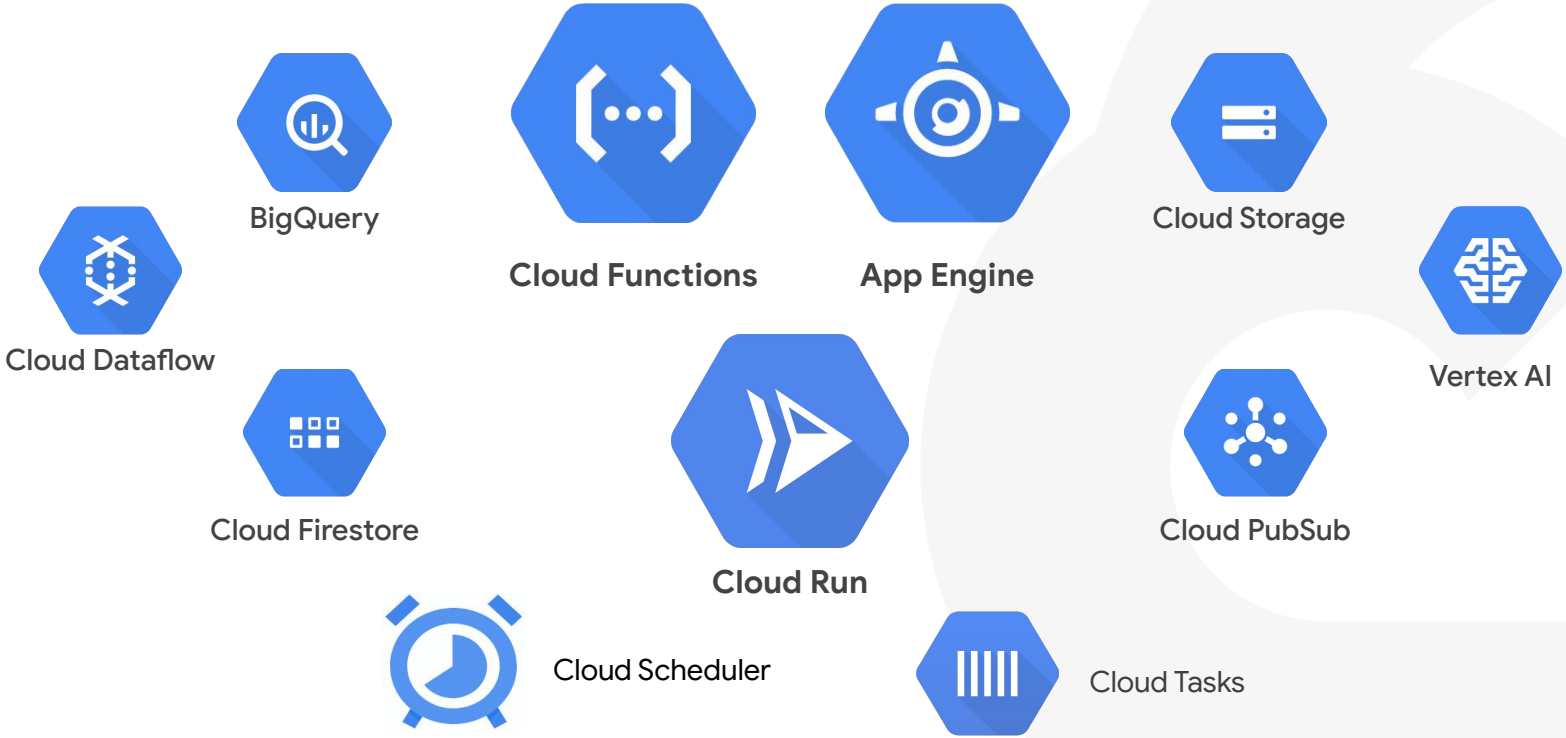
How to go Serverless.

Analytics & Machine Learning

Google Cloud

Serverless on GCP

(for Databases, Data Processing, and Machine Learning)



Cloud Functions

Cloud Functions

Event-driven serverless compute platform

React to events

HTTP
Firestore
Pub/Sub
Cloud Scheduler
Cloud Storage
Firebase

Serverless

No servers to manage
Scale up fast
Scale down to zero
No patches/updates
Pay for exact usage

Choice of runtimes

Node.js 10, 12, 14, 16
Python 3.7, 3.8, 3.9
Go 1.11, 1.13, 1.16
Java 11
Ruby 2.7
PHP 7.4
.NET Core 3.1

Google Cloud

Cloud Functions

Event-driven serverless compute platform

Longer Processing Times PREVIEW

6x execution time
from 9 min to 60 min (HTTP Only)

Longer Data processing pipelines

Longer Machine Learning pipelines

Larger Upgraded Instances

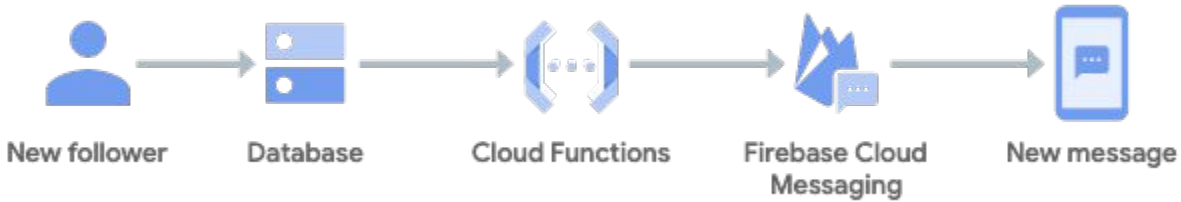
256 MB
512 MB
1024 MB
2048 MB
4096 MB, 2 vCPU
8GB, 2 vCPU
16GB, 4 vCPU NEW

Google C

Use Case: Integration with third-party services and APIs

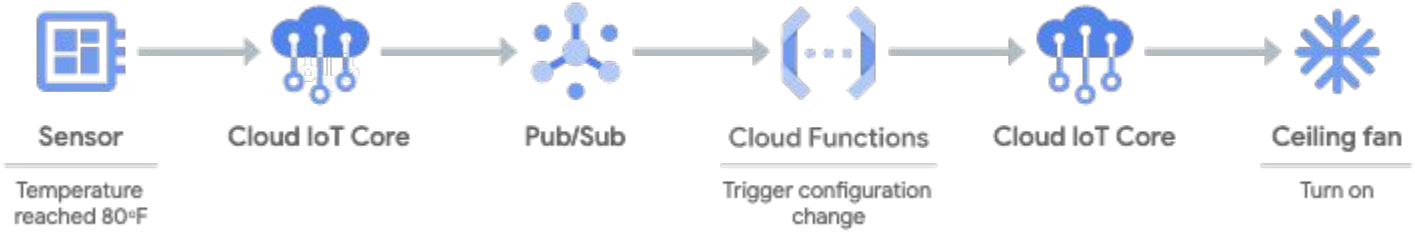
Google Cloud

Use Case: Serverless mobile back ends



Google Cloud

Use Case: Serverless IoT back ends

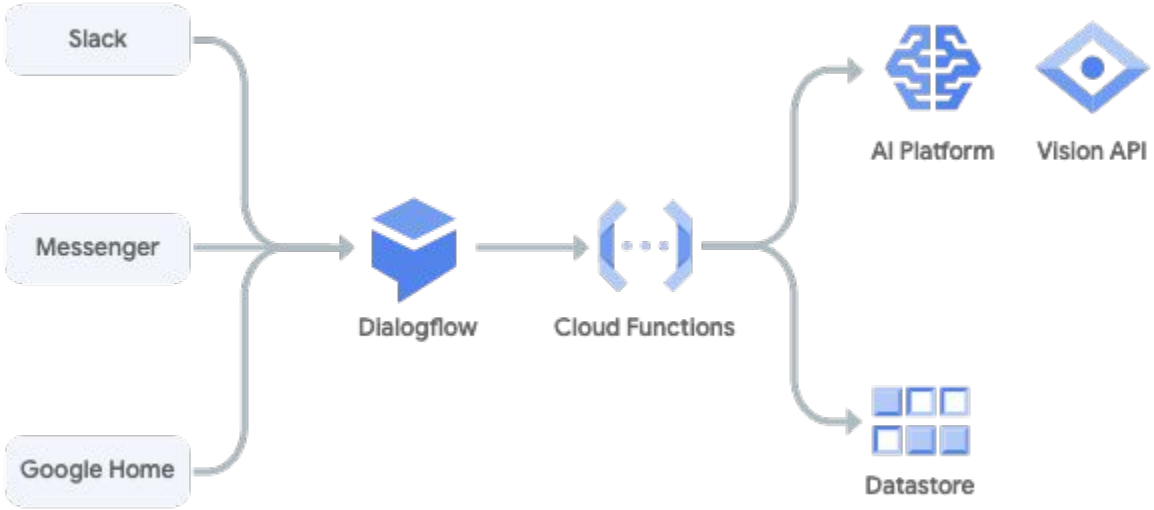


Google Cloud

Use Case: Real-time file processing

Google Cloud

Use Case: Virtual assistants and conversational experiences



Google Cloud

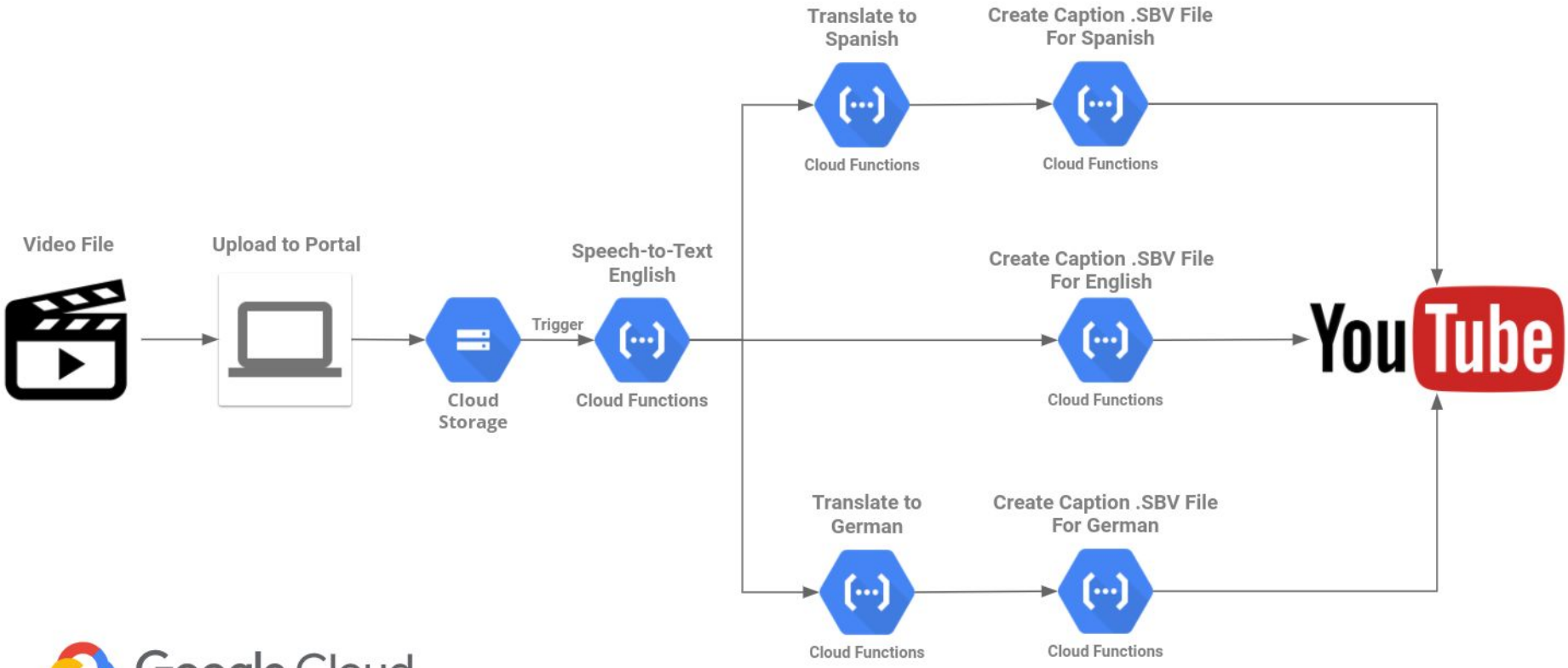
In Class Bonus: Make an REST POST API call to my deployed Cloud Function. Here are the details of the API (Hint: You can use CURL or the python requests library)

Endpoint URL:
https://us-east1-stealth-air-23412.cloudfunctions.net/ztest1

Header:
{'content-type': 'application/json'}

Payload:
{'name': 'ENTER_YOUR_NAME_HERE'}

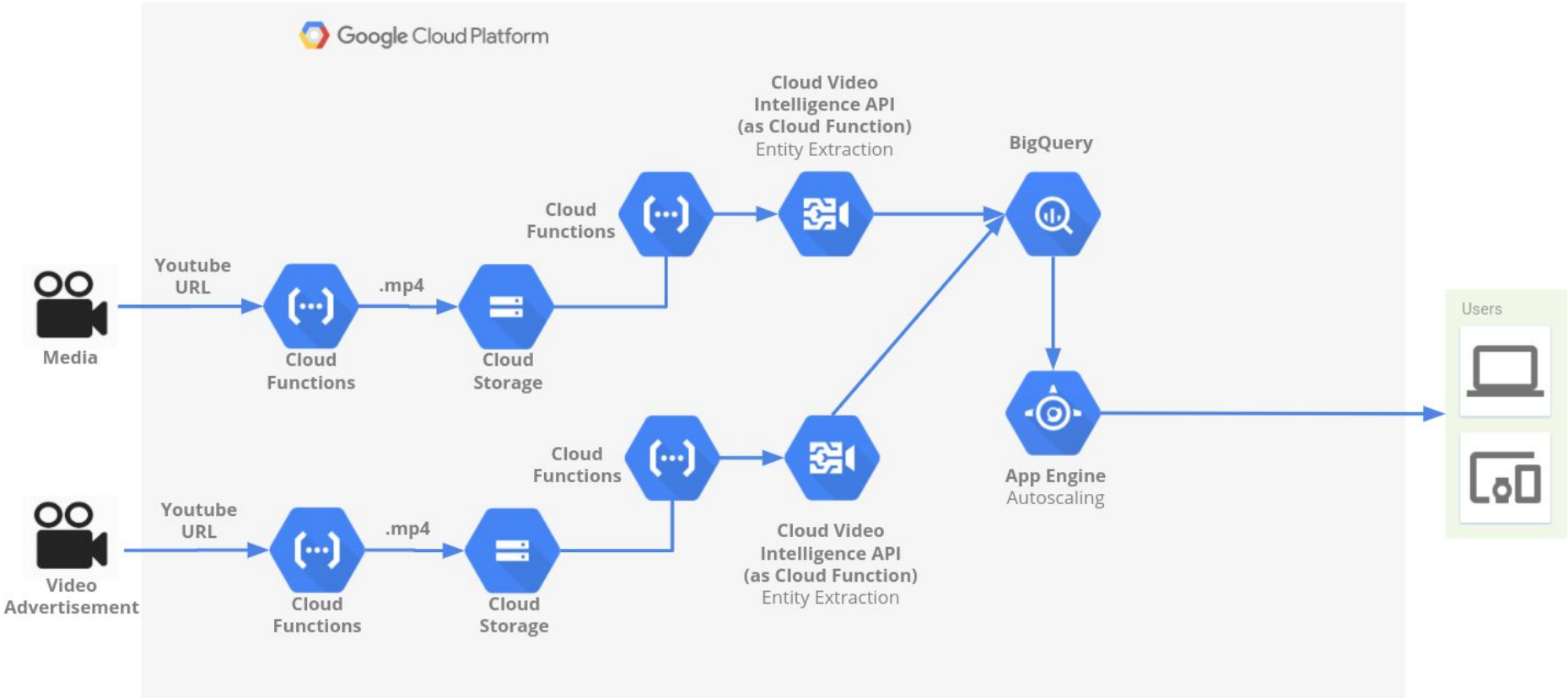
Process Flow



Google Cloud

Solution Architecture

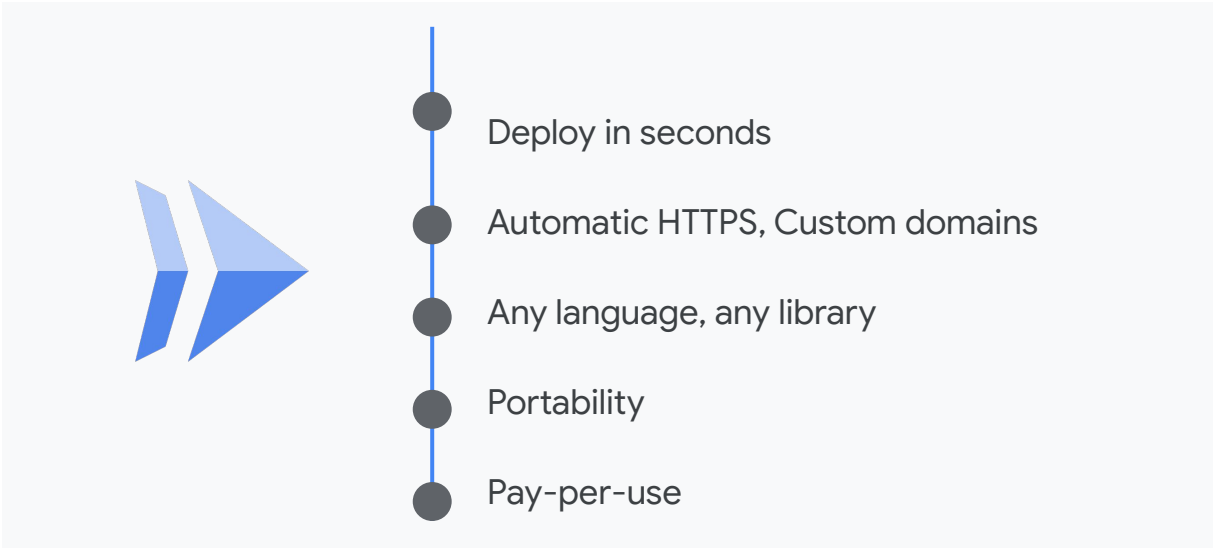
Google Cloud Architecture



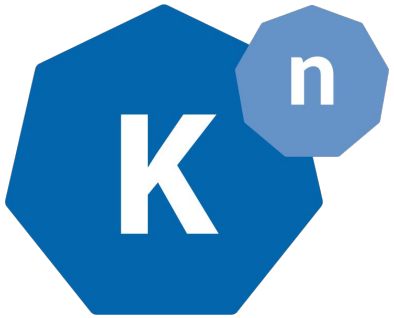
Cloud Run

Cloud Run

Run stateless **containers** on a **fully managed** environment



Google Cloud



Knative

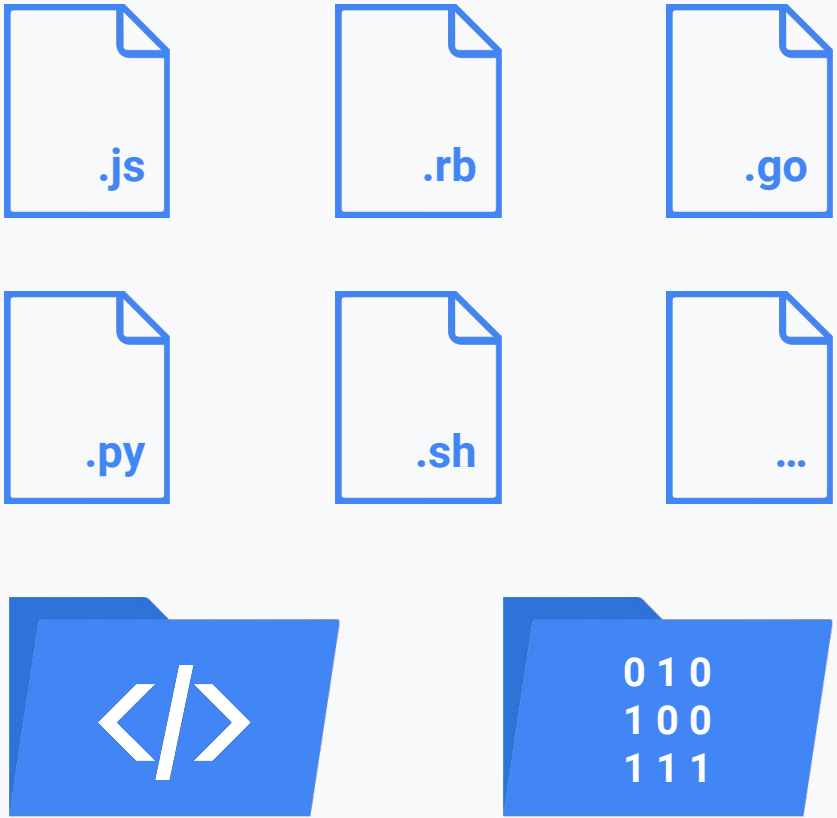
open source **building blocks**

for serverless on Kubernetes

Google Cloud

Containers

- Any Language
- Any Library
- Any Binary



Google Cloud

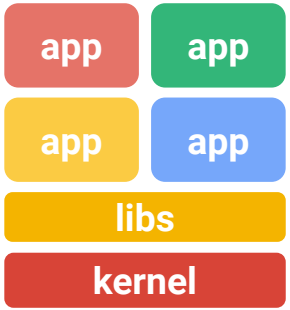
Detour to Containers…

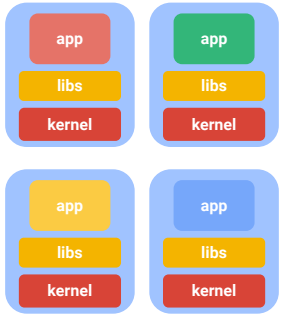
What are containers?

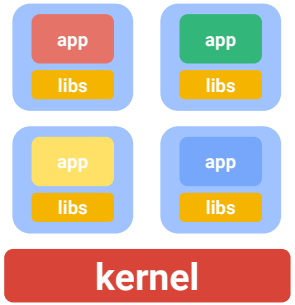
- Containers are a **method of packaging** an application executable and its dependencies (runtime, system tools, system libraries, configuration),
and **running the package as a set of resource-isolated processes**

- **Buzzwords associated with containers**
  - Lightweight
  - Portable/Standard
  - Productivity
  - Secure

From physical servers to containers







Physical Machine

- ✗ No isolation
- ✗ Common libs
- ✗ Highly coupled apps and OS

VMs

- ✓ Isolation
- ✓ No common libs
- ✗ Expensive and inefficient
- ✗ Hard to manage

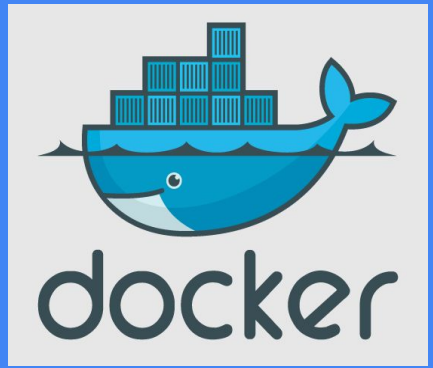
Containers

- ✓ Isolation
- ✓ No common libs
- ✓ Less overhead
- ✗ Less dependency on host OS

Google Cloud

Docker

- Dominant container tool
- Released OSS in 2013 by DotCloud Inc. (hosted PaaS)
- **Made it easy and fast to create and run container images**
- Spread like wildfire
- DotCloud Inc. → Docker Inc. (sold the PaaS business)
- Started to move up the stack in 2015 as revenue pressure grew



Everything at Google runs in containers

- Gmail, Web Search, Maps, ...
- MapReduce, batch, ...
- GFS, Colossus, ...
- Even **Google's Cloud Platform:** Our VMs run in containers

We launch over **4 billion** containers **per week**

Google Cloud

Back to Cloud Run

Deploy from local source

Single command to Build, Push and Deploy local sources.

Using GCP Buildpacks, Cloud Build and Artifact Registry

Supports:

- Go 1.10+
- Node.js 10+
- Python 3.7+
- Java 8 & 11
- .NET Core 3.1+
- Dockerfile

```
$ ls
go.mod       main.go

$ gcloud run deploy
Deploying from source. To deploy a container use [--image].

Building using Buildpacks and deploying container to Cloud Run
service [my-service] in project [my-project] region [europe-west1]

⠶ Building and deploying...
  ✓ Uploading sources...
  ⠶ Building Container...
  . Creating Revision...
  . Routing traffic...
```

Gradual rollouts & Rollbacks

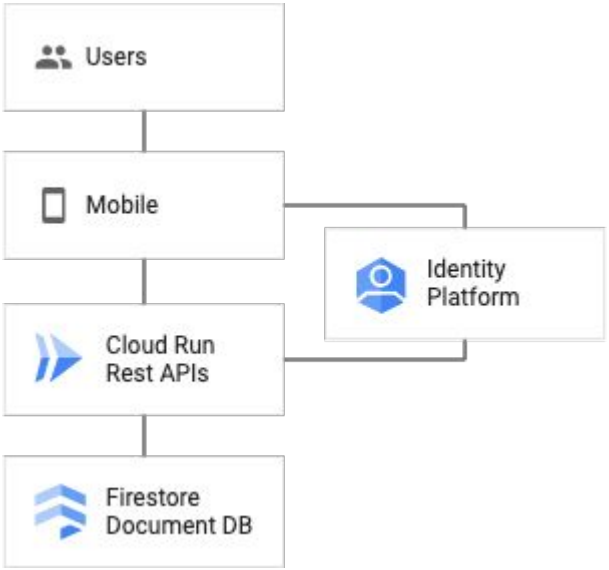
Specify % traffic between revisions

Blue / Green deployments

Get URLs for specific revisions

```
# Gradual rollout

$ gcloud beta run deploy myservice \
                    --image gcr.io/project/image:f5bd774 \
                    --no-traffic \
                    --tag green
$ curl https://green---myservice-12345-us.a.run.app


$ gcloud beta run services update-traffic myservice \
                    --to-tags green=1

$ gcloud beta run services update-traffic myservice \
                    --to-tags green=10

$ gcloud beta run services update-traffic myservice \
                    --to-tags green=50

$ gcloud beta run services update-traffic myservice \
                    --to-tags green=100


# Rollback

$ gcloud run services update-traffic myservice
                    --to-revisions my-service-0002-joy=100
```

Use Case: Web services: REST APIs backend



Cloud Run Demo

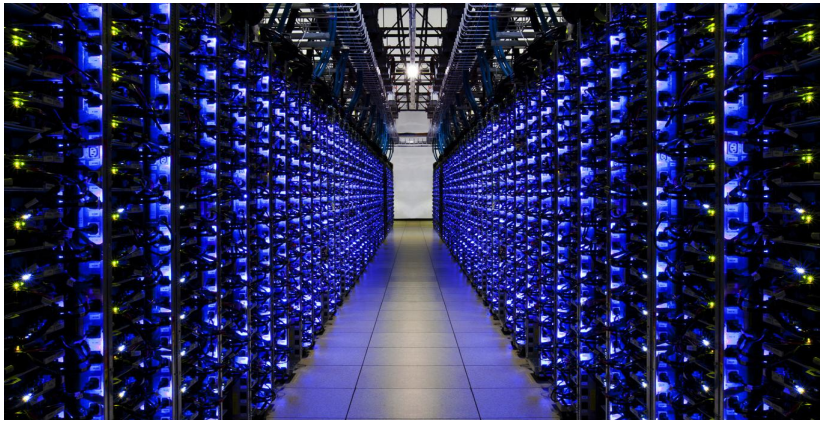
Kubernetes

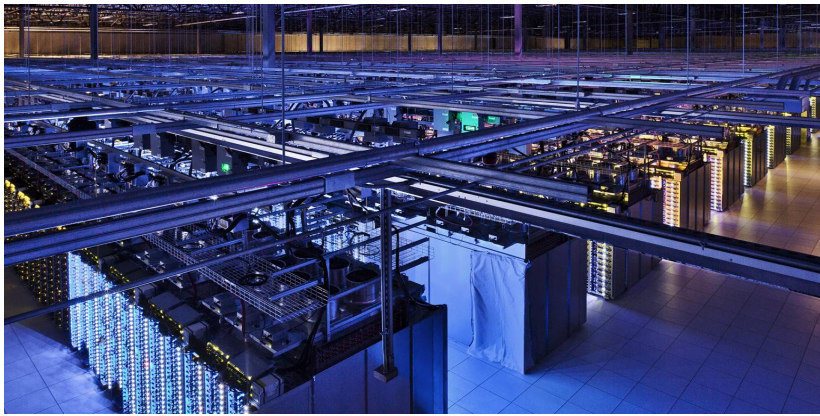
Back to Containers…

Deploying containers at scale is different!

A **fundamentally different** way of managing applications requires different tooling and abstractions

- Deployment
- Management, monitoring
- Isolation
- Updates
- Discovery
- Scaling, replication, sets





Google Cloud

Kubernetes

Greek for *“Helmsman”*; also the root of the words *“governor”* and *“cybernetic”*

- Manages container clusters
- Inspired and informed by Google’s experiences and internal systems
- Supports multiple cloud and bare-metal environments
- Supports multiple container runtimes
- **100% open source**, written in Go

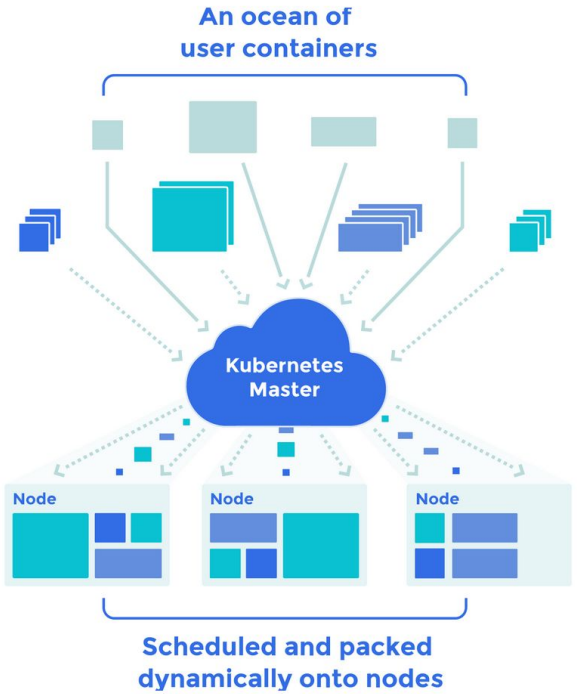
Manage applications, not machines

In simple terms...

Think of Kubernetes as the OS for your compute fleet

It provides features similar to an OS for a host:

- Scheduling workload
- Finding the right host to fit your workload
- Monitoring health of the workload
- Scaling it up and down as needed
- Moving it around as needed



Google Cloud

Simplify container management with Kubernetes

Declarative management

Portable platform

Infrastructure abstraction

Google Cloud

Kubernetes handles...

Scheduling:
Decide what pods to run on which nodes

Lifecycle and health:
Keep my containers running despite failures

Scaling:
Make sets of containers bigger or smaller

Naming and discovery:
Find where my containers are now

Load balancing:
Distribute traffic across a set of containers

Storage volumes:
Provide data to containers

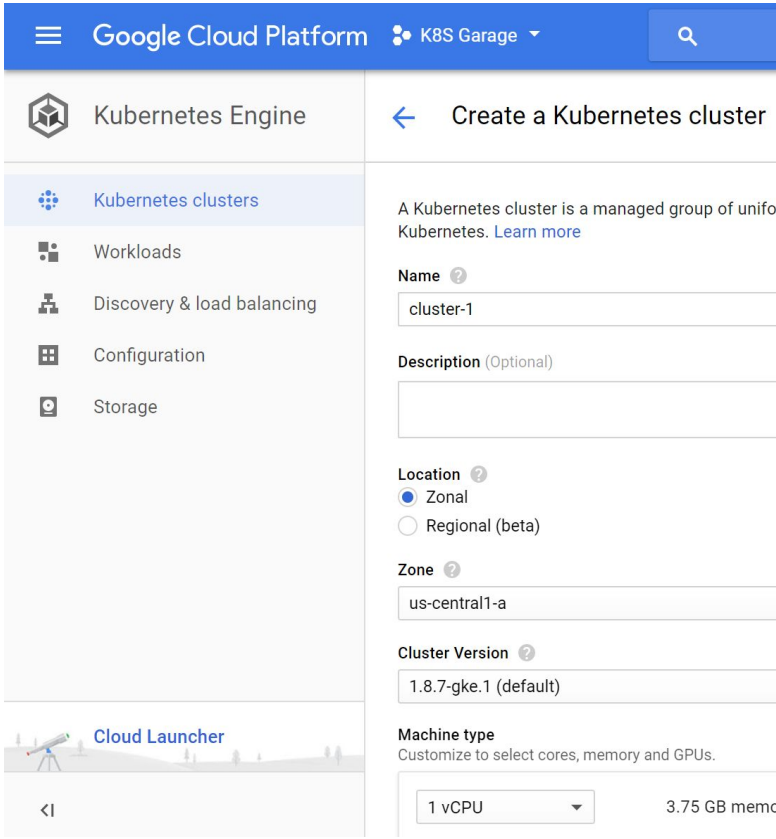
Logging and monitoring:
Track what's happening with my containers

Debugging and introspection:
Enter or attach to containers

Identity and authorization:
Control who can do things to my containers

Kubernetes the easy way

- Start a cluster with one-click
- View your clusters and workloads in a single pane of glass
- Google keeps your cluster up and running



Google Cloud

Thank you!!

Open discussion and Q+A

Thanks!

Advanced Big Data:
Google Cloud & Serverless

Dan Zaratsian
AI/ML Solutions Architect, Gaming @ Google