Build Your Own Serverless With Knative

Alex Gervais

ConFoo Montreal

Feb. 27, 2020

Bonjour-Hi!

Outdoorsy, data-driven, eternal student, not so geeky creative mind and traveler. Alex is a curious, **introverted** and humble character. Working by day as a **Senior Software Developer** at **Datawire** he has many years of savoir-faire building full-stack systems from cloud infrastructures to backend services and DevOps tools. Alex is a **Kubernetes** early adopter who thrives on collaboration and contributor to many **Cloud-Native** projects.

@alex_gervais on **twitter**
alexgervais on **github**

Table of Contents

Landscape

- Paradigms
- Cloud vendor solutions
- Runtime environment
- Abstractions

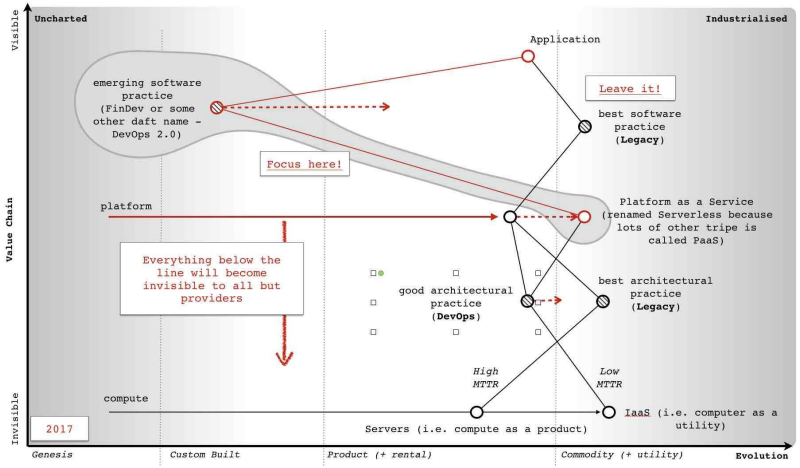
Serverless

- Event-driven architectures
- Pay-per-use
- Servicefull

Strategic Planning

Simon Wardley argues, **serverless** changes computing into a **commodity** like electricity. [3]

Required reading: Wardley Maps



Use cases

As a **developer**…

- Glue functionalities
- Smaller microservices
- High-volume functions

Use cases

As an **operator**…

- Provide the fundamentals of a self-service platform
- Use spare capacity in existing infrastructure
- Examine the internals

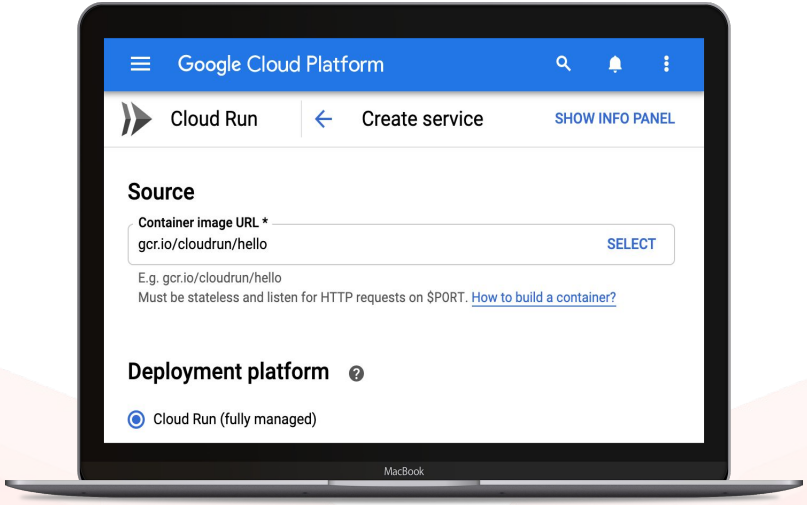
Functions

~~Functions~~

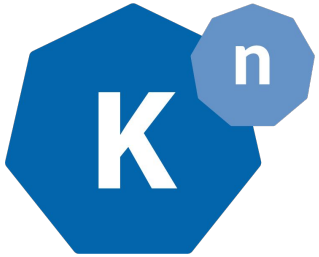
Containers

Google Cloud Run

- Fully managed
- Pick your infrastructure
- Loose container runtime contract



Knative



- 2 700+ ★ github.com/knative
- Latest stable release: v0.12.1
- v0.1.0 in July 2018

- The platform that powers Cloud Run
- Open source
- Knative on any cloud

1. Knative Build
2. Knative Serving
3. Knative Eventing

1. Knative Build

~~1. Knative Build~~

Build system that is designed to address a common need for cloud native development.

Deprecated.

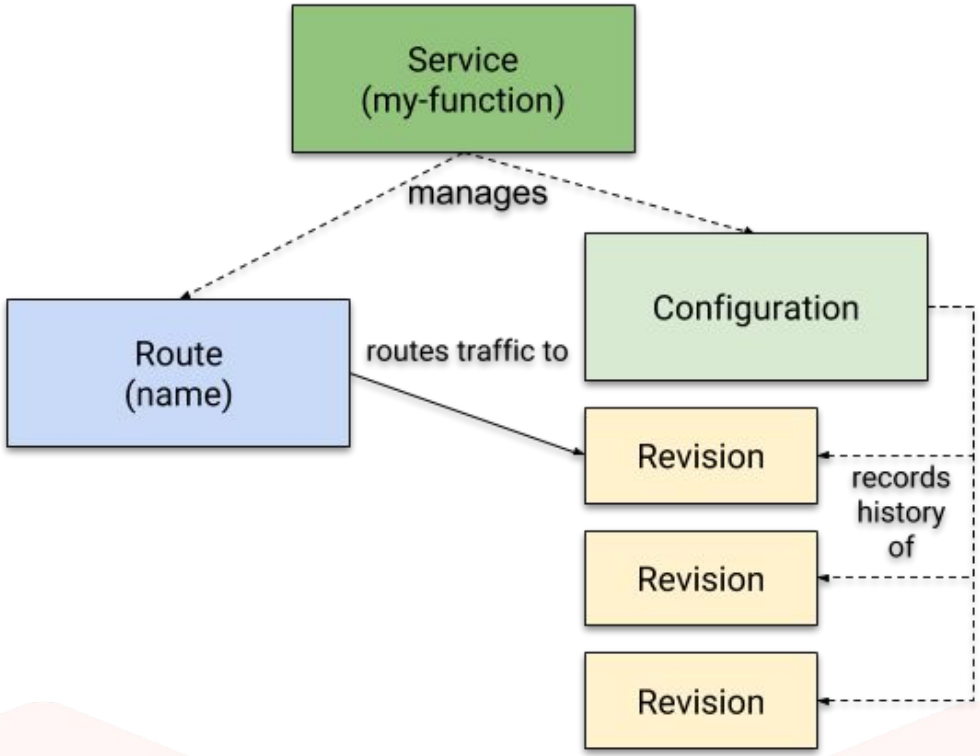
2. Knative Serving

2. Knative Serving

Serving provides scale-to-zero, request-driven compute functionality. Essentially the **execution** and **scaling** components of a serverless platform.[2]

Alternative to traditional Kubernetes svc, deployments, and hpa.

2. Knative Serving Resources

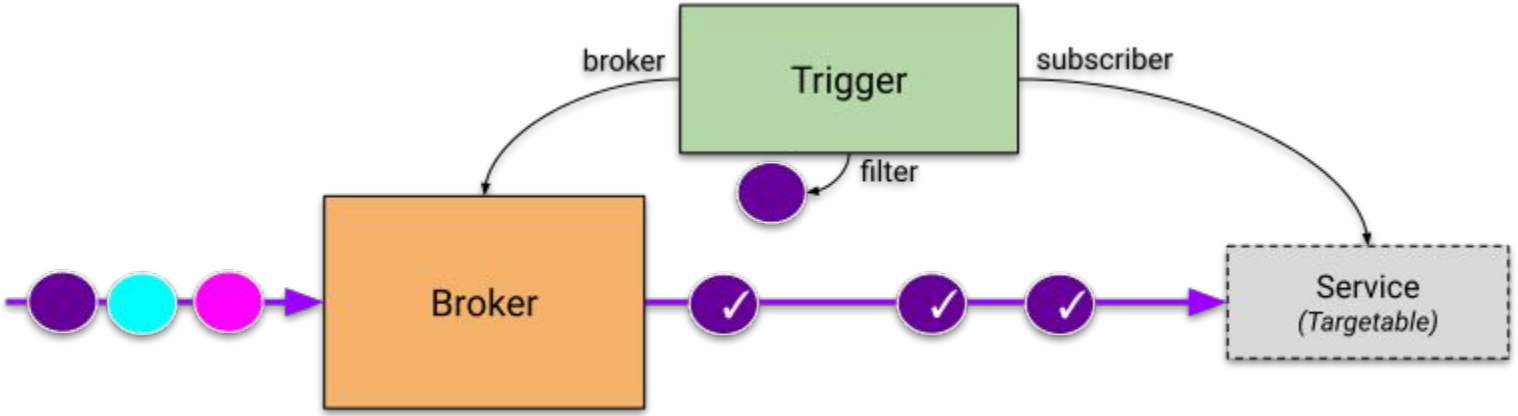


3. Knative Eventing

3. Knative Eventing

Eventing provides abstracted **delivery** and **subscription** mechanisms which allow for building loosely-coupled and event-driven serverless applications.[2]

3. Knative Eventing Resources



3. Knative Eventing Channels

Because the Broker requires a persistence layer...

GCP PubSub	**InMemoryChannel**
KafkaChannel	NatssChannel

3. Knative Eventing Sources

The Broker can handle events from multiple sources of interest

AWS SQS	BitBucket	Google Cloud Scheduler
Apache Camel	Cron Job	Kubernetes API
Apache Kafka	GCP PubSub	...

CloudEvents

- 1 600+ ★ github.com/cloudevents/spec
- Latest stable release: v1.0
- v0.1 in April 2018

- A **specification** for describing event data in a common way.
- Ensure cross-service interoperability.
- Knative Eventing is consistent with the CloudEvents spec.

CloudEvent serialized as JSON

```
{
    "specversion" : "1.0",
    "type" : "com.github.pull.create",
    "source" : "https://github.com/cloudevents/spec/pull",
    "subject" : "123",
    "id" : "A234-1234-1234",
    "time" : "2018-04-05T17:31:00Z",
    "comexampleextension1" : "value",
    "comexampleothervalue" : 5,
    "datacontenttype" : "text/xml",
    "data" : "<much wow=\"xml\"/>"
}
```

Let’s serve some HTTP

HTTP Events

Knative depends on an Ingress/Gateway which is capable of **routing** HTTP requests to Knative Services.

service mess /ˈsərvəs mes/ noun 1. the result of spending more compute resources than your actual business logic dynamically generating and distributing Envoy proxy configs and TLS certificates.

Kelsey Hightower @kelseyhightower
5:43pm - 13 Jul 2019

The Ambassador Edge Stack

- 2 500+ ★ github.com/datawire/ambassador
- Latest stable release: v1.2.0
- v0.1.3 in April 2017

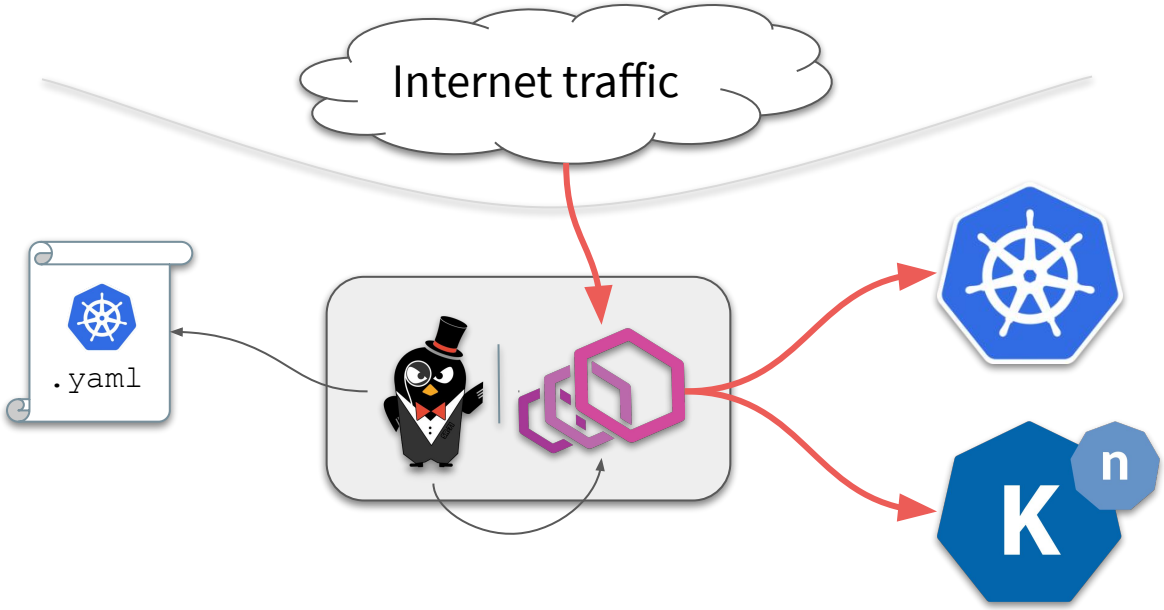
- API Gateway built on **Envoy**
- Open source
- Designed for Kubernetes
- Lives at the network edge to manage ingress traffic

Ambassador & Knative

Installing Knative with Ambassador gives us an **alternative** to installing a service mesh for routing to applications with the Knative Serving project.

Note that Istio is required for the Knative Eventing project.

Ambassador & Knative



Demo time!

Installation

Requires a Kubernetes cluster v1.15 or newer.

```
$ kubectl apply -f \
https://github.com/knative/serving/releases/download/v0.11.0/serving-crds.yaml

$ kubectl apply -f \
https://github.com/knative/serving/releases/download/v0.11.0/serving-core.yaml

$ kubectl apply -f \
https://getambassador.io/yaml/ambassador/ambassador-knative.yaml

$ kubectl apply -f \
https://getambassador.io/yaml/ambassador/ambassador-service.yaml
```

Deploy an application.yaml

```
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: helloworld-go
  namespace: default
spec:
  template:
    spec:
      containers:
        - image: gcr.io/knative-samples/helloworld-go
          env:
            - name: TARGET
              value: Go Sample v1
```

Voilà!

```
$ kubectl get ksvc helloworld-go
NAME            URL                                        LATESTCREATED
helloworld-go   http://helloworld-go.default.example.com
helloworld-go-xkpfs


$ curl -v -H "Host: helloworld-go.default.example.com" <ambassador-ip>:80

< HTTP/1.1 200 OK
< content-length: 20
< content-type: text/plain; charset=utf-8
< x-envoy-upstream-service-time: 16
< server: envoy
<
Hello Go Sample v1!
```

End Result

Lightweight portable cloud-agnostic serverless platform.

Limitations

- Cold start
- Concurrency
- Resource consumption

Thanks!

Credits

[1] How Knative Can Unite Kubernetes and Serverless

[2] Knative: What is it and why should you care?

[3] Serverless Challenges We Need to Overcome

[4] Cloud Run

[5] Tutorial: Cloud-Agnostic Serverless - Sebastien Goasguen, TriggerMesh

[6] Installing Knative with Ambassador

[7] Self-Serverless: Why Run Knative Functions on Your Kubernetes Cluster?

[8] Knative Presentation from OSCON 2019 Portland, OR