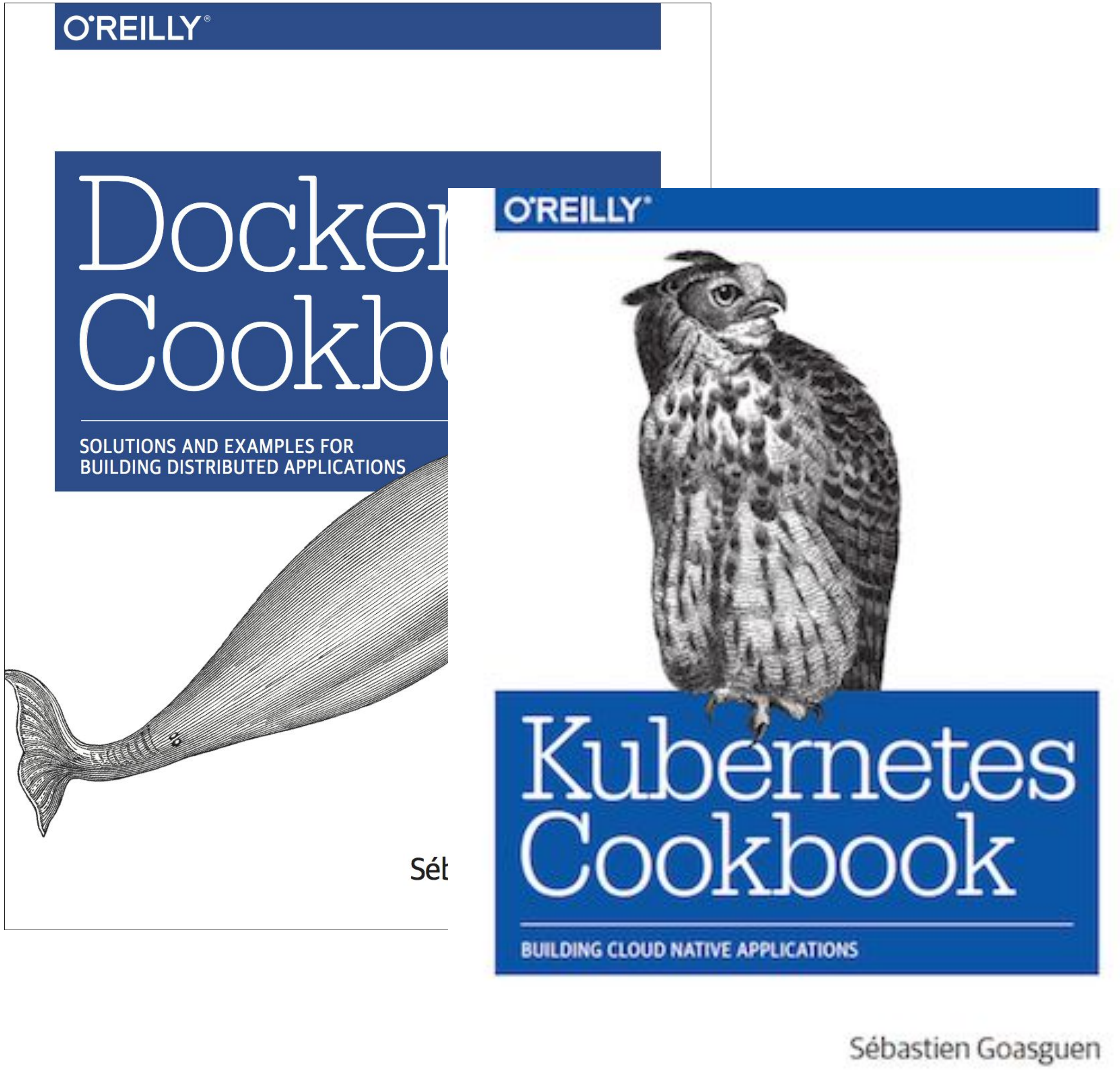
Packaging Applications in a Serverless World

- Sebastien Goasguen @sebgoa

Hi,

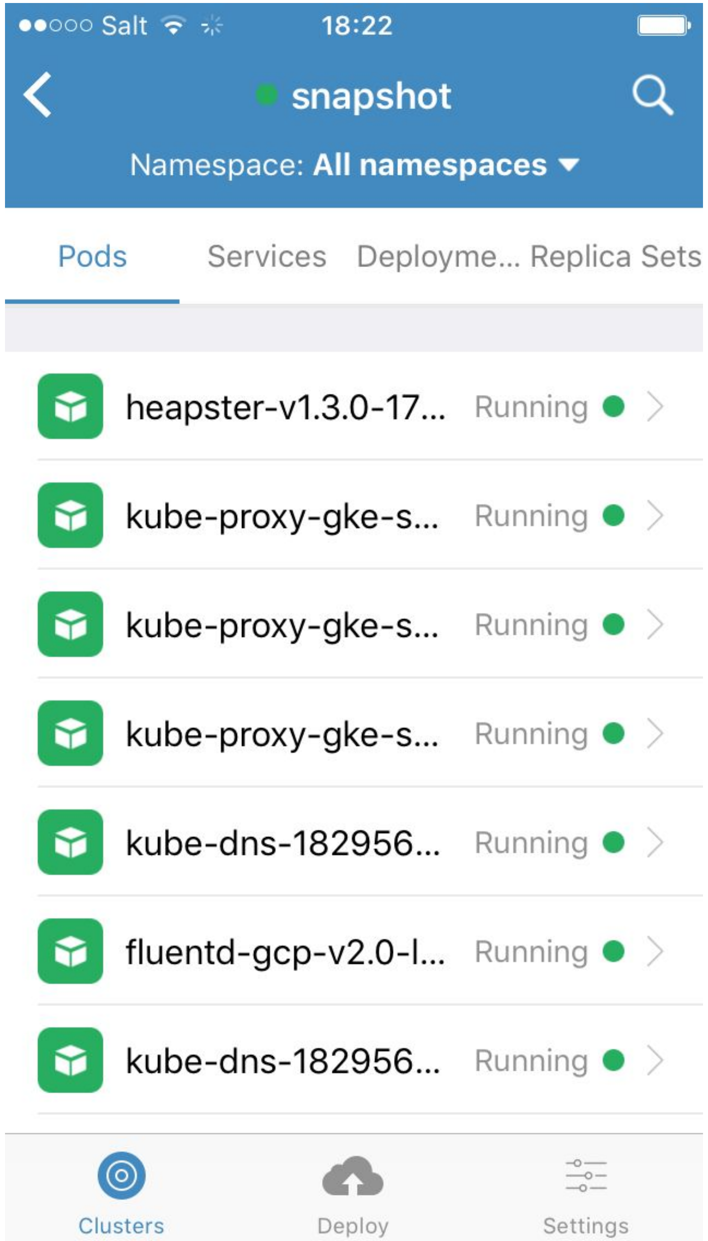
- 14 years in US, long time in Academia
- Electromagnetics and Nanoelectronics
- Open Science Grid, TeraGrid, CERN
- System Experimentators
- Advocate / Speaker / Writer
- Product Owner / Product Manager


O'REILLY
Docker Cookbo
SOLUTIONS AND EXAMPLES FOR BUILDING DISTRIBUTED APPLICATIONS
Séb
O'REILLY
Kubernetes Cookbook
BUILDING CLOUD NATIVE APPLICATIONS
Sébastien Goasguen

K8s work

- kompose (second project in incubator after helm)

- kmachine
- Cabin
- **kubeless (github.com/kubeless/kubeless)**
- O’Reilly training, LF training for certification …



Namespace: All namespaces

heapster-v1.3.0-17... Running ●

kube-proxy-gke-s... Running ●

kube-proxy-gke-s... Running ●

kube-proxy-gke-s... Running ●

kube-dns-182956... Running ●

fluentd-gcp-v2.0-l... Running ●

kube-dns-182956... Running ●

Serverless: A few basics to warmup

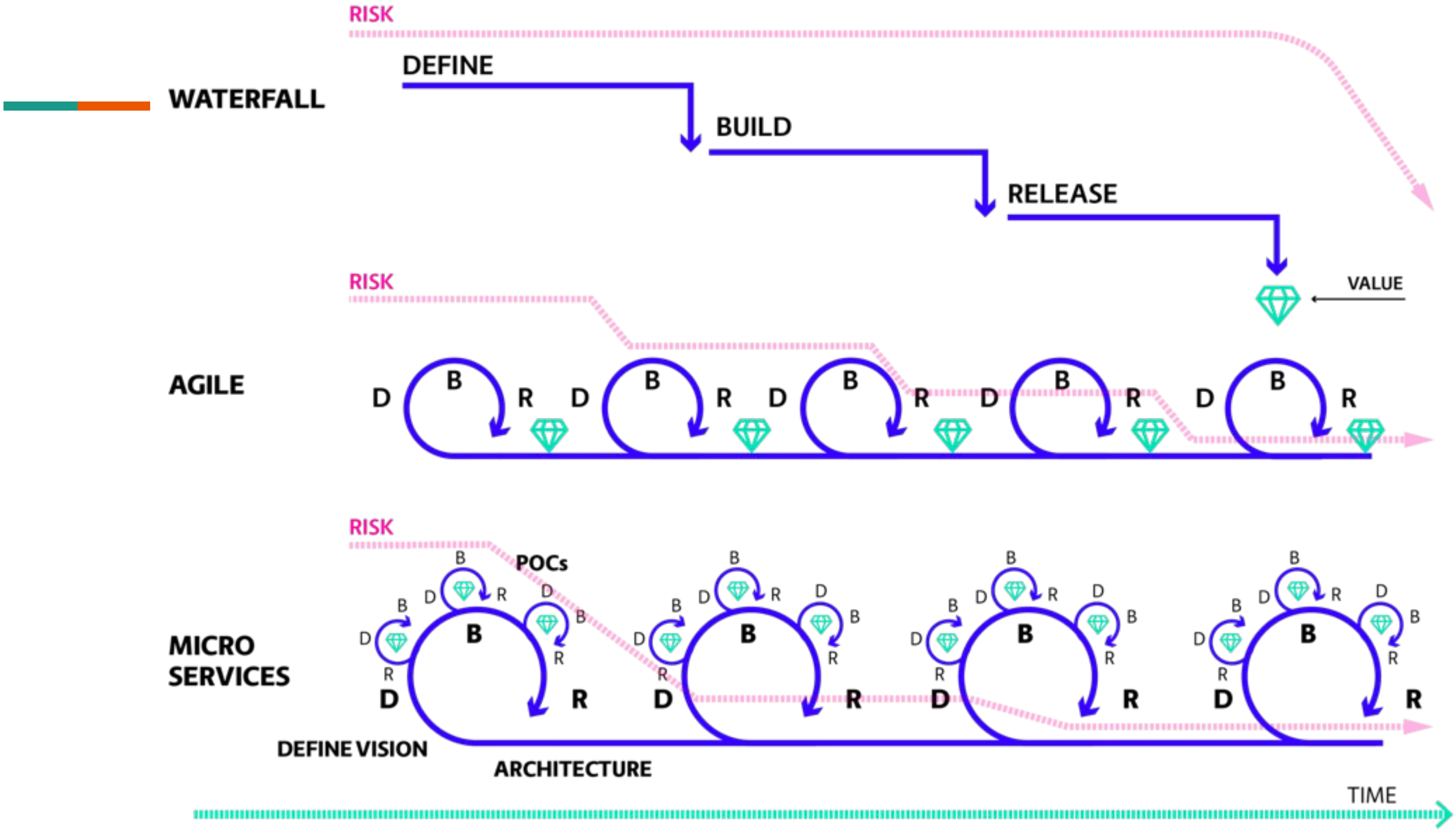
Preface

no silver bullet

an evolution

that we must pay attention to, understand it

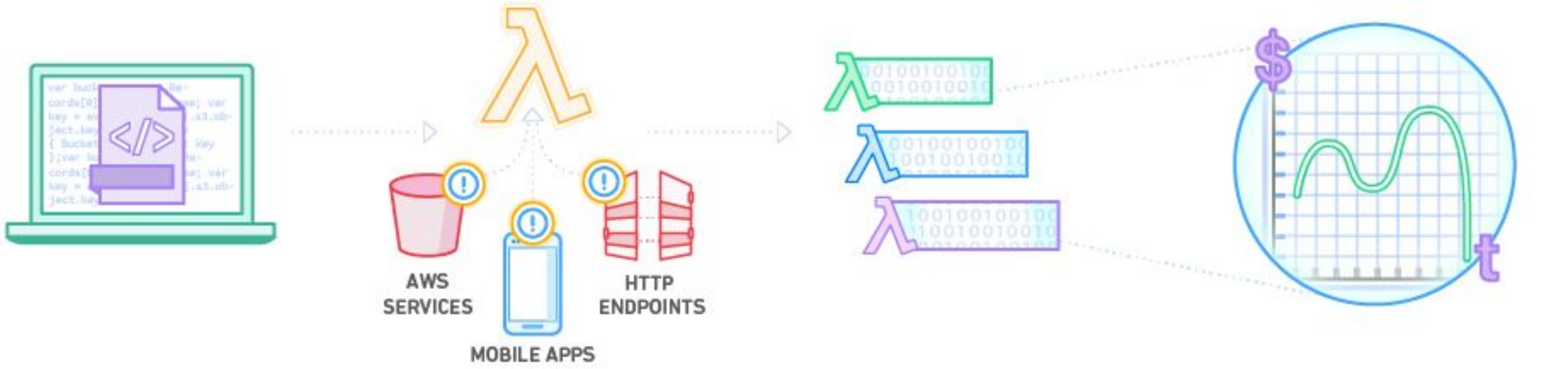
ease of application deployment -> dev onboarding


RISK
DEFINE
WATERFALL
BUILD
RELEASE
VALUE
RISK
AGILE
D
B
R
D
B
R
D
B
R
D
B
R
D
B
R
RISK
POCs
MICRO SERVICES
B
D
R
B
D
R
D
B
R
D
B
R
DEFINE VISION
ARCHITECTURE
TIME

Serverless

There is no 'serverless'
it's just someone elses fully managed
execution environment that I only pay a fraction
of a cent for whenever my function is run

AWS Lambda



Upload your code to AWS Lambda

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

Lambda runs your code only when triggered, using only the compute resources needed

Pay just for the compute time you use

AWS CLI

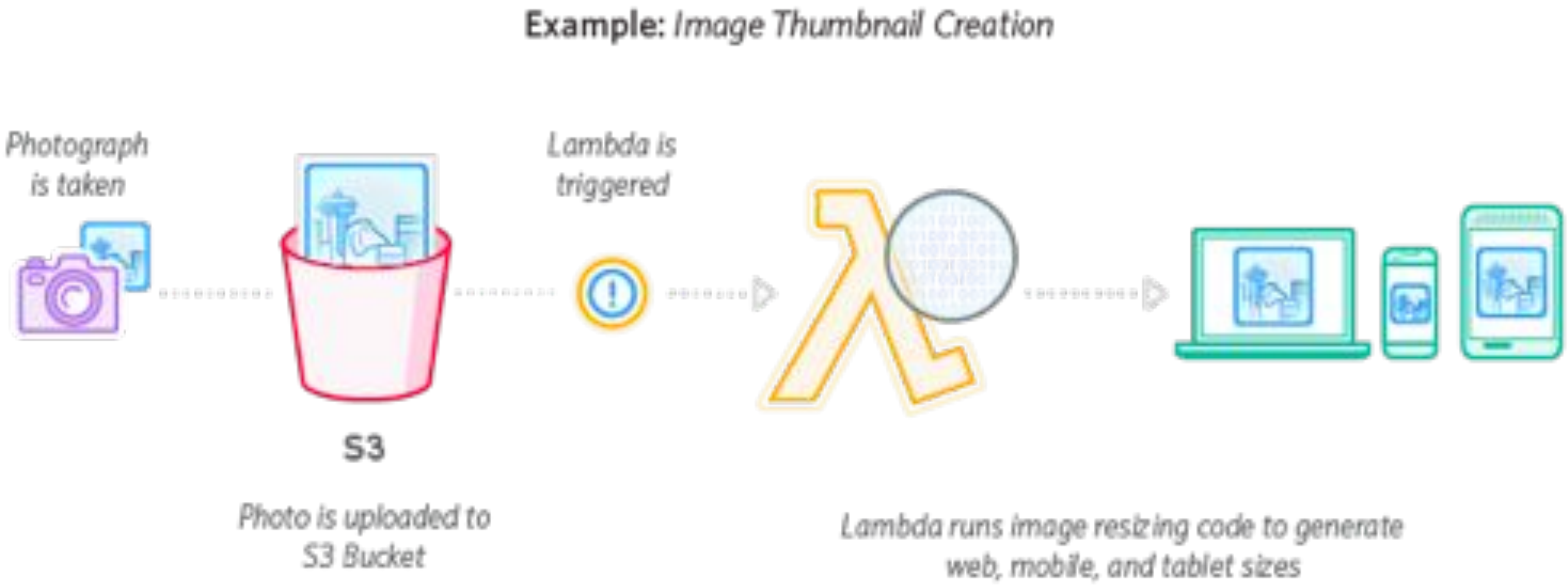
```
$ aws lambda create-function \
--region us-west-2 \
--function-name CreateThumbnail \
--zip-file fileb://file-path/CreateThumbnail.zip \
--role role-arn \
--handler CreateThumbnail.handler \
--runtime runtime \
--profile adminuser \
--timeout 10 \
--memory-size 1024
```

What is a serverless application ?

“service full”

“finer pay as you use model”

What type of Apps ?

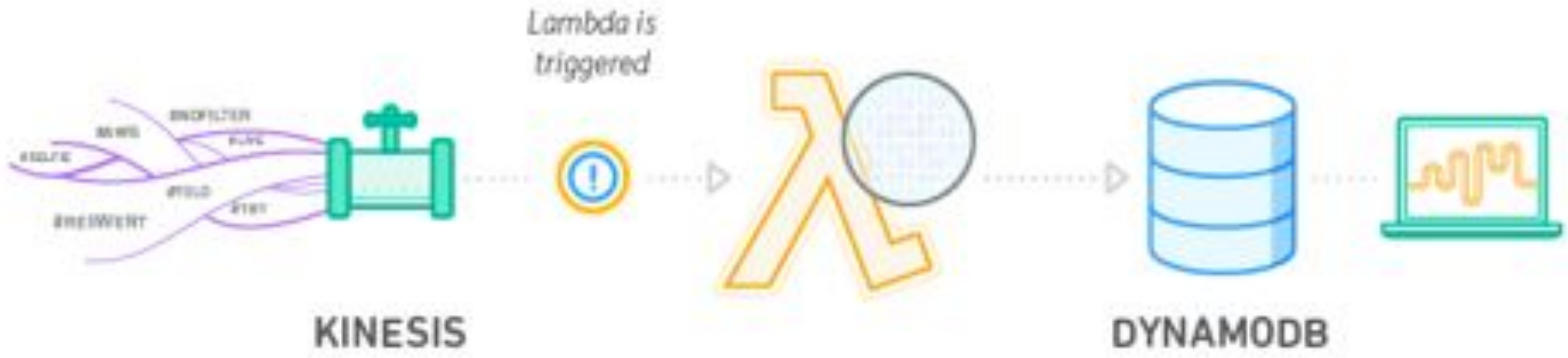


Example: *Image Thumbnail Creation*

Photograph is taken
S3
Photo is uploaded to S3 Bucket
Lambda is triggered
Lambda runs image resizing code to generate web, mobile, and tablet sizes

Data Streams and Processing

Example: *Analysis of Streaming Social Media Data*



Lambda is triggered

Social media trend data immediately available for business users to query

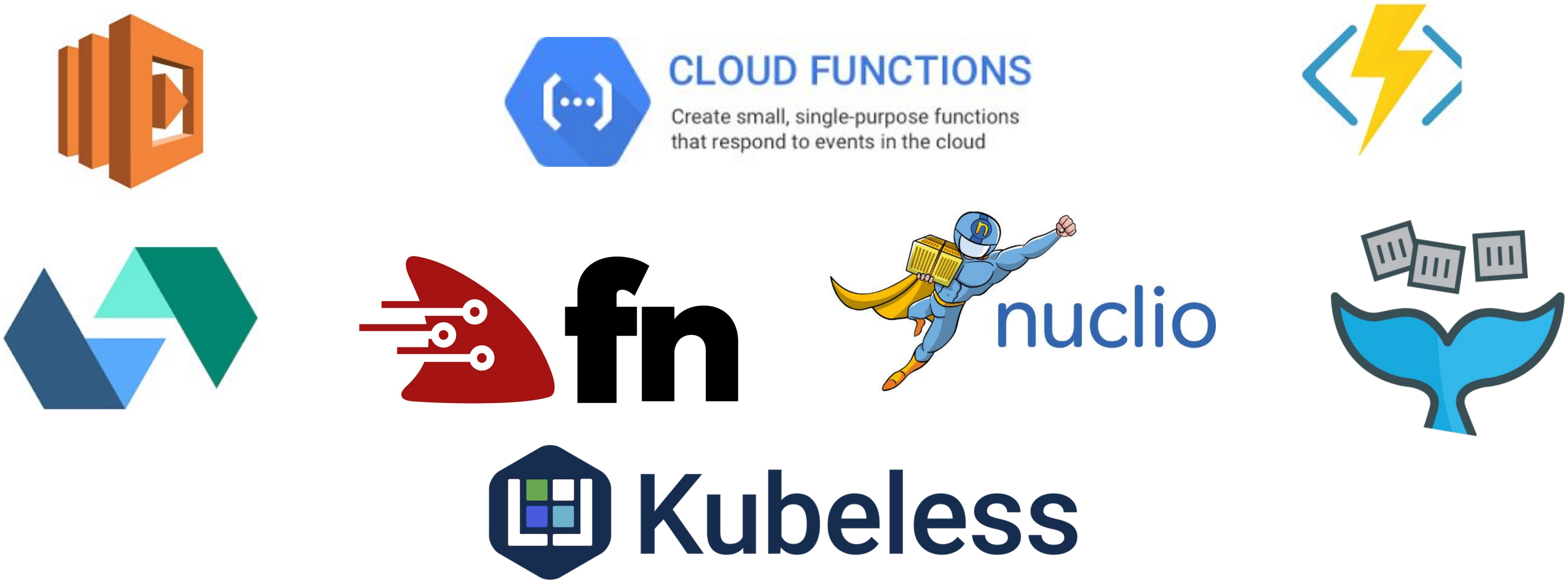
Social media stream is loaded into Kinesis in real-time

Lambda runs code that generates hashtag trend data and stores it in DynamoDB

Concepts

- Function endpoints (CGI ?)
- Triggers
- Events

Serverless and FaaS Solutions


CLOUD FUNCTIONS
Create small, single-purpose functions that respond to events in the cloud
fn
nuclio
Kubeless

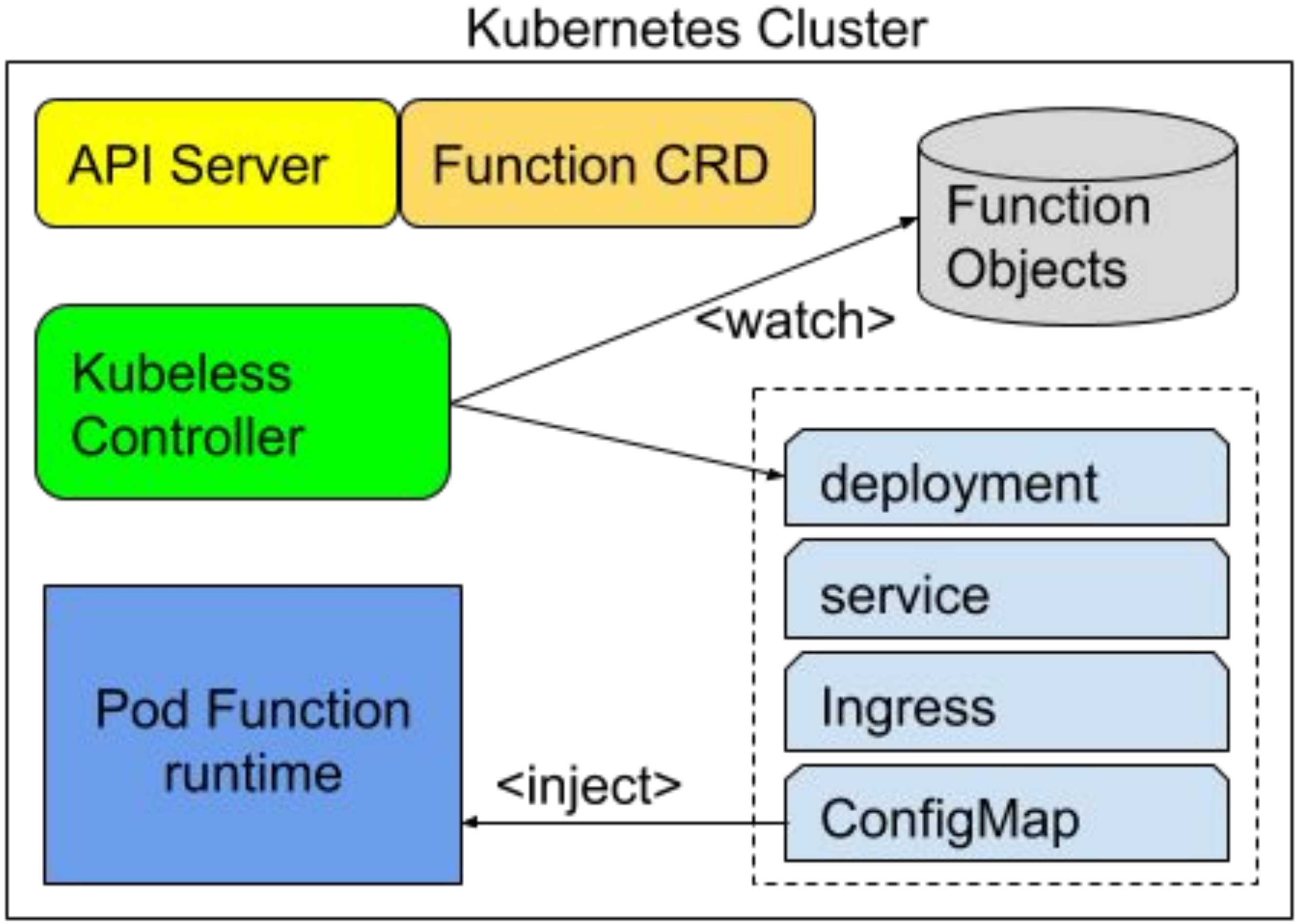
Building Kubeless

- FaaS on on-premises, does it even make sense ?

Kubernetes Native

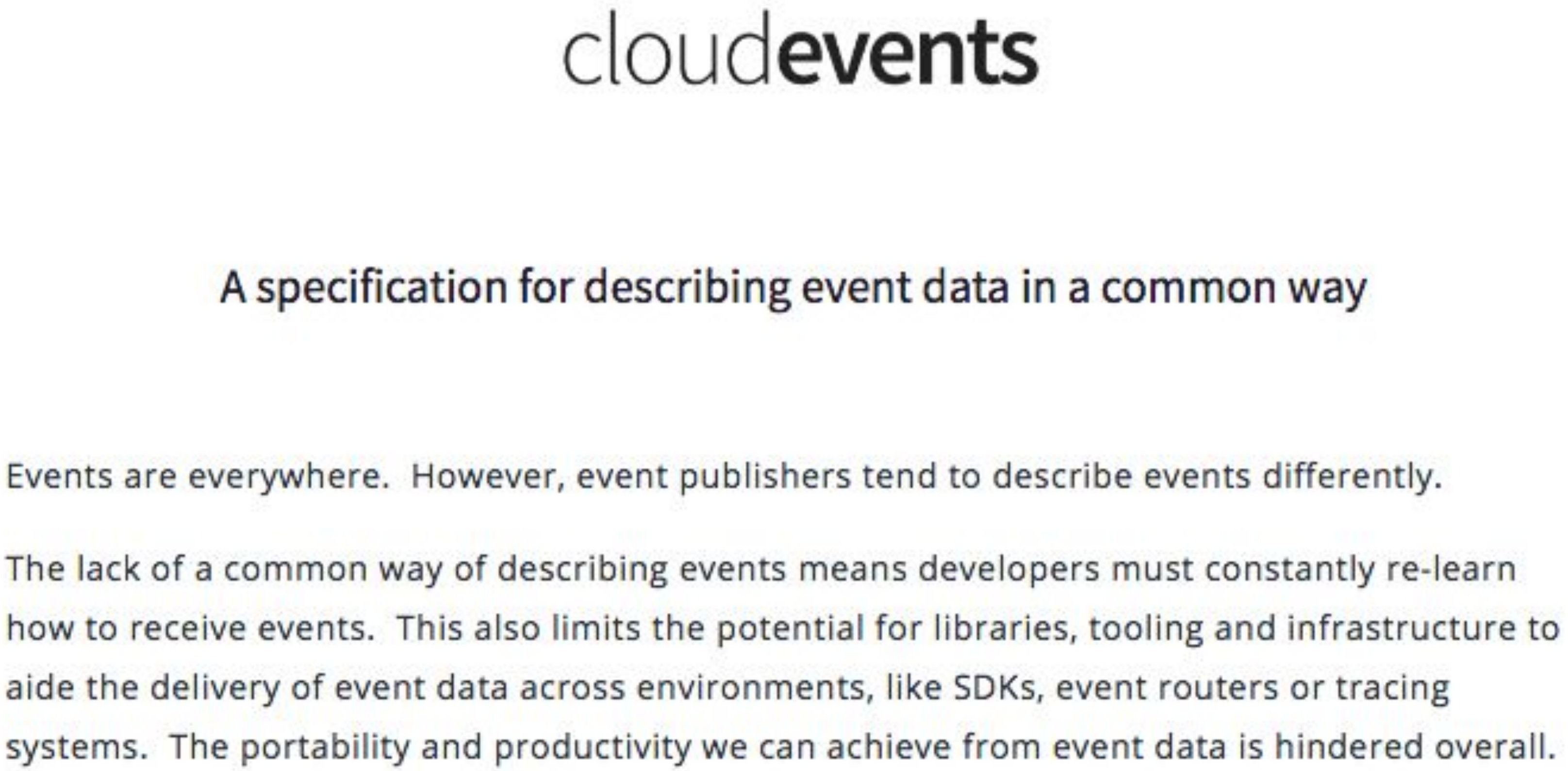
- Extend Kubernetes
- Use Kubernetes API Objects
- Leverage other primitives like: Horizontal Pod AutoScaler
- Use CNCF monitoring - Prometheus
- Use Istio/Envoy for traffic encryption, distributed tracing and more

Architecture



But this is all about Events and Triggers !

CloudEvents



cloudevents

A specification for describing event data in a common way

Events are everywhere. However, event publishers tend to describe events differently.

The lack of a common way of describing events means developers must constantly re-learn how to receive events. This also limits the potential for libraries, tooling and infrastructure to aide the delivery of event data across environments, like SDKs, event routers or tracing systems. The portability and productivity we can achieve from event data is hindered overall.

Decoupling Runtimes and Triggers

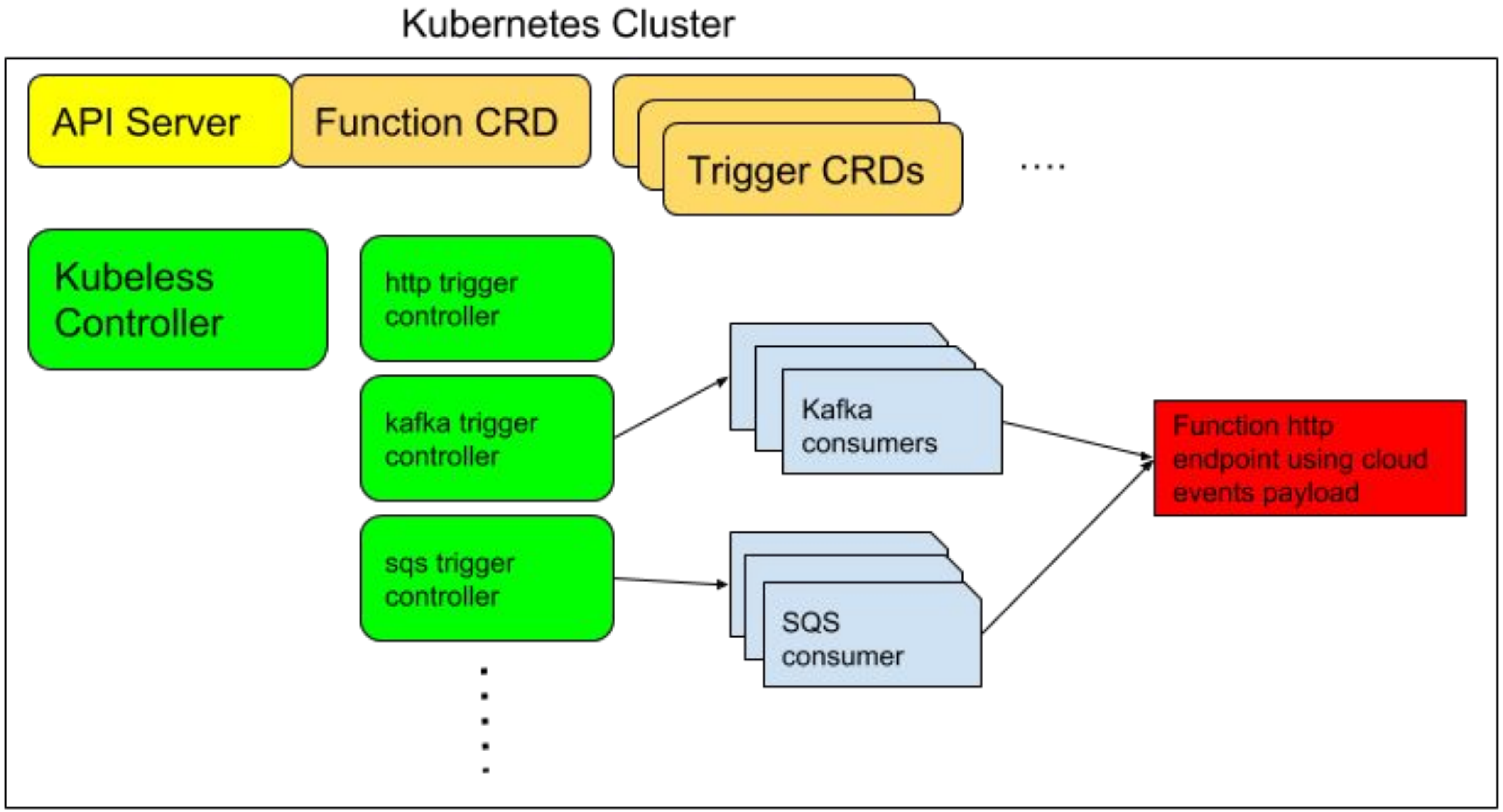
To be able to scale languages and event sources

Break up the event consuming mechanism from the runtime

All function are now called over HTTP

“Trigger” controllers spawns consumers and dispatches to endpoints.

Scaling Event Sources



Triggers (aka binding) as CRD

```
apiVersion: kubeless.io/v1beta1
kind: KafkaTrigger
metadata:
  labels:
    created-by: kubeless
    function: slack
  name: slack
spec:
  functionSelector:
    matchLabels:
      created-by: kubeless
      function: slack
  topic: slack
```

CI/CD and Packaging

Closely tied to how you package a k8s app

Docker as a package

Function/code inside the image, onus on the users to build docker

Where are we on the spectrum ? (k8s or PaaS or something else)

Reproducibility (base image ?) and immutability issues (tags ?)

Applications on k8s

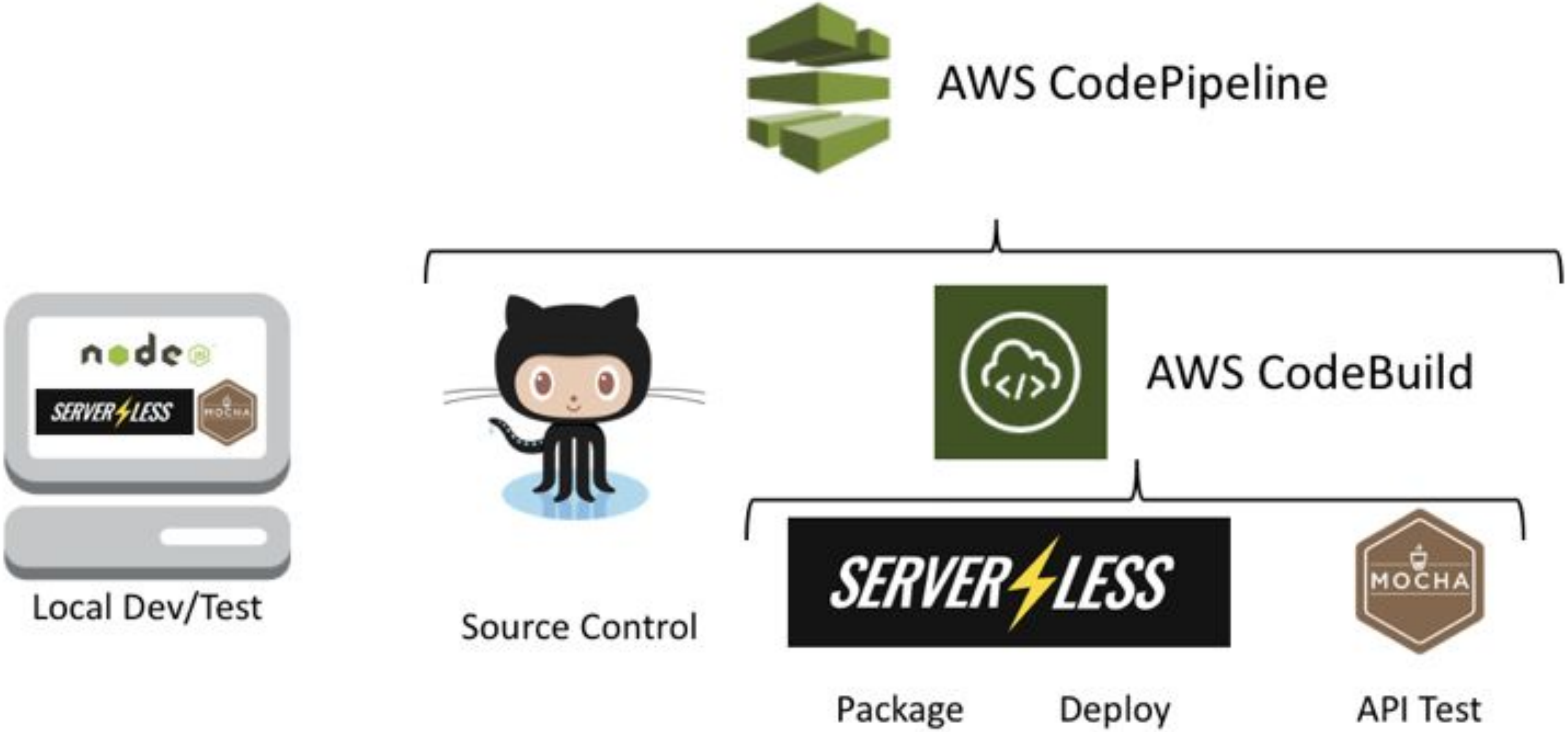
Multiple containers

Lots of other _things_ (services, ingress, pvc, cm ...)

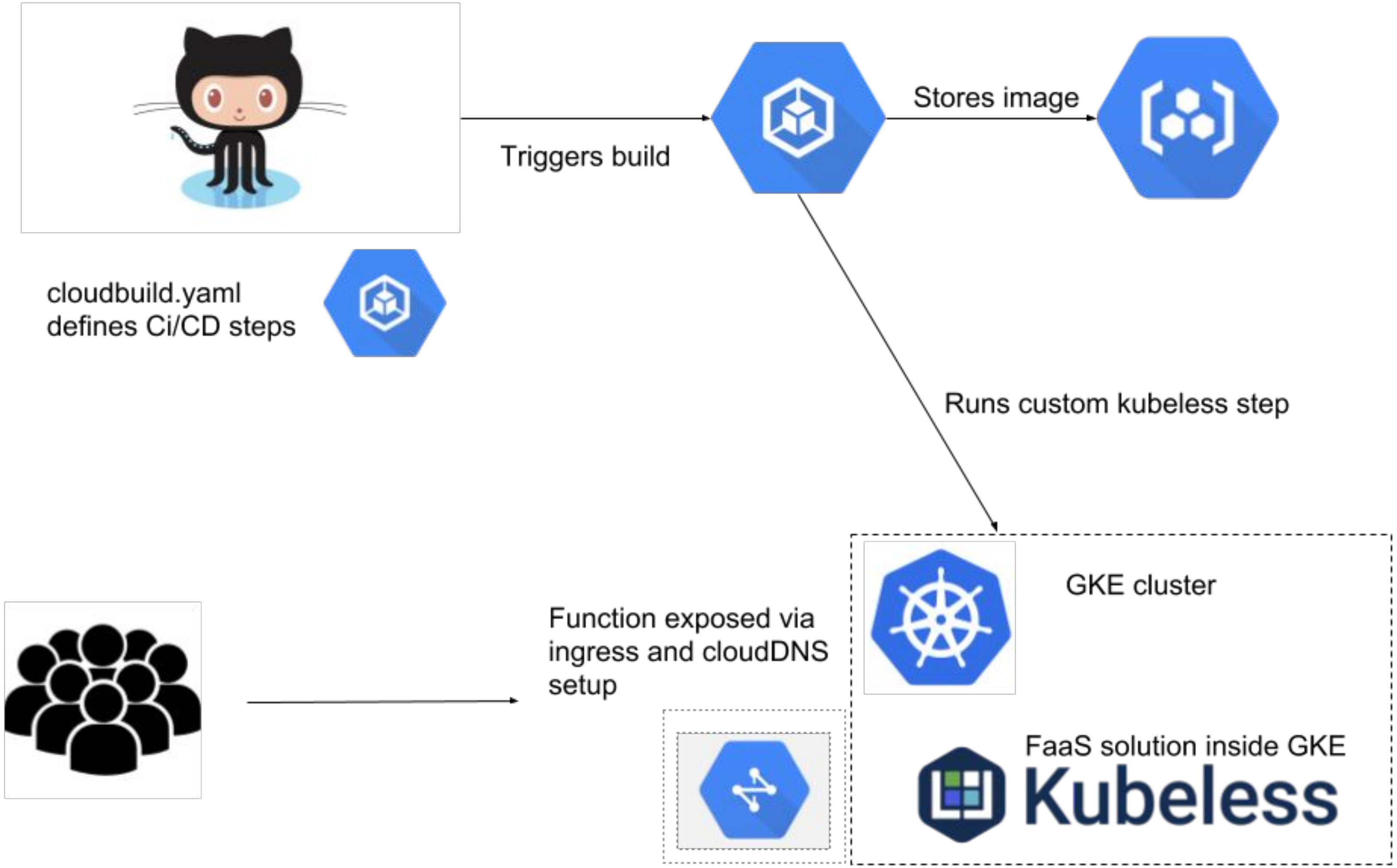
60 configuration tools

Helm as the leading package manager (chart as a tarball of k8s manifests)

CI/CD on AWS


AWS CodePipeline
node
SERVERLESS
MOCHA
Local Dev/Test
Source Control
AWS CodeBuild
SERVERLESS
MOCHA
Package
Deploy
API Test

CI/CD on GCP/GKE



Triggers build

Stores image

cloudbuild.yaml
defines Ci/CD steps

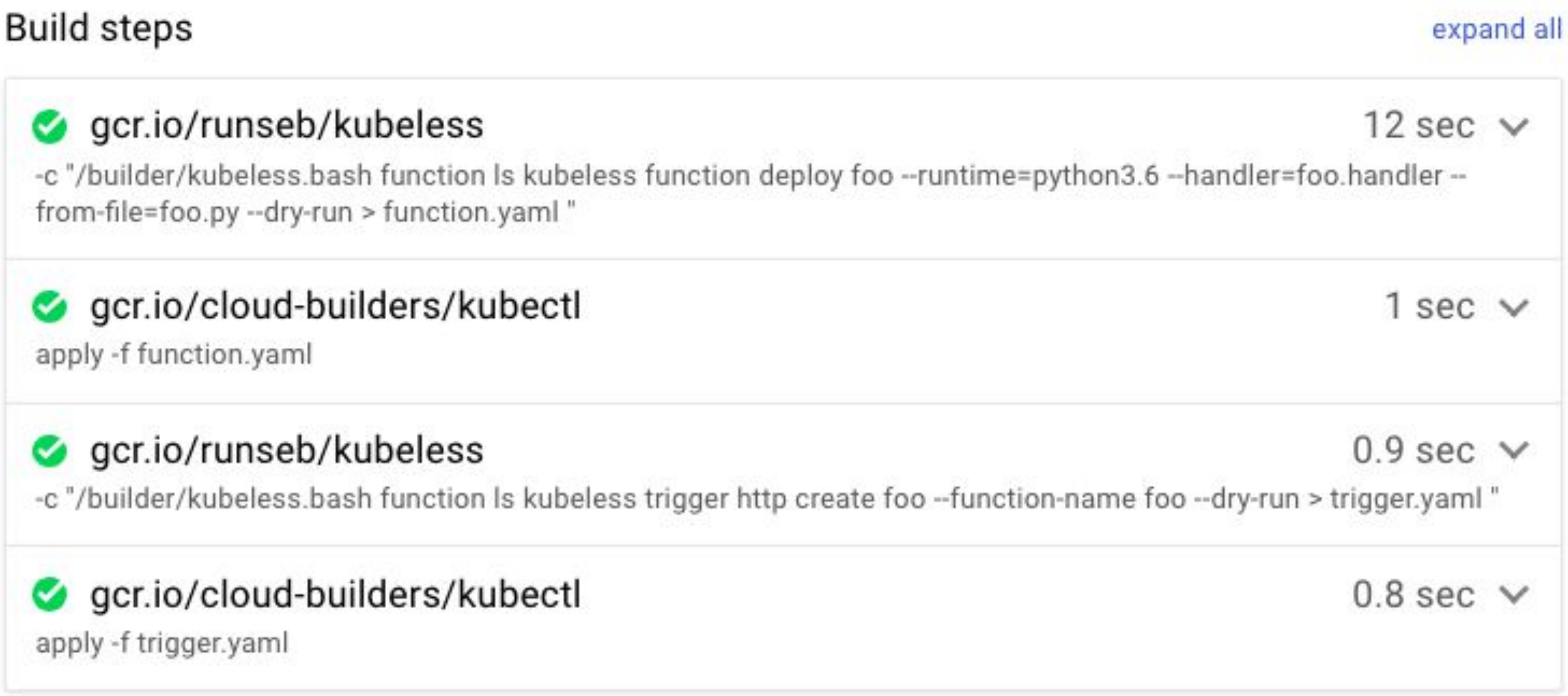
Runs custom kubeless step

GKE cluster

Function exposed via
ingress and cloudDNS
setup

FaaS solution inside GKE

Google Container Builder



Build steps

✓ gcr.io/runseb/kubeless 12 sec

```
-c "/builder/kubeless.bash function ls kubeless function deploy foo --runtime=python3.6 --handler=foo.handler --from-file=foo.py --dry-run > function.yaml "
```

✓ gcr.io/cloud-builders/kubectl 1 sec

```
apply -f function.yaml
```

✓ gcr.io/runseb/kubeless 0.9 sec

```
-c "/builder/kubeless.bash function ls kubeless trigger http create foo --function-name foo --dry-run > trigger.yaml "
```

✓ gcr.io/cloud-builders/kubectl 0.8 sec

```
apply -f trigger.yaml
```

No package

Users do not write Dockerfile

Users do not write k8s manifest

There is no artifact (at least in the eye of the beholder)

There is just code

… and a declarative manifest

Loving it !

Not infra focused

Not build focused

Code focused

but **Delivery focused**

Service Full App Package

https://github.com/bitnami-labs/redisdemo

A tarball / zip of declarative manifests

Includes functions

Includes cloud services

```
helm install --name todo . --set ingress.host=todo.kubeless.sh
```

Serverless is coming

Still very early

Cloud vs On-premises

CI/CD in the Cloud

A blob is a blob is a blob

Immutable and reproducible

Thank You
@sebgoa
https://github.com/kubeless/kubeless