Cloud Native New Horizon Technology

Michael Maximilien (aka Dr. Max)
CTO for OSS development
International Business International

Building Cloud Native Application with Knative and Community Update

19 May 2020

IBM Cloud

cloud native applications

- **what is cloud native?**
  - “modern” web applications and APIs
  - follows (more or less) 12factor.net principles
  - stateless, image-based, external configuration, container deployed and managed, scale up horizontally

- **example cloud native platforms?**
  - CF —Cloud Foundry is a complete PaaS solution
  - Kubernetes (Kube) — flexible OSS container orchestrator
  - Other — Heroku, Mesos, Docker Swarm, and homegrown

IBM **Cloud**

@maximilien

serverless

- **what is serverless?**
  - pay-as-you use computing
  - dynamic allocation of required resources for execution
  - functional decomposition of your application's features
  - event-driven computing — react to events for execution flow
- **advantages?**
  - pay only for what you use (?)
  - scale when needed, and down to 0, when no use
  - consume events expressly in application code and config
- **drawbacks —** may require
  - re-architecture of app or at least some features
  - new management of services (functions)

IBM **Cloud**

@maximilien

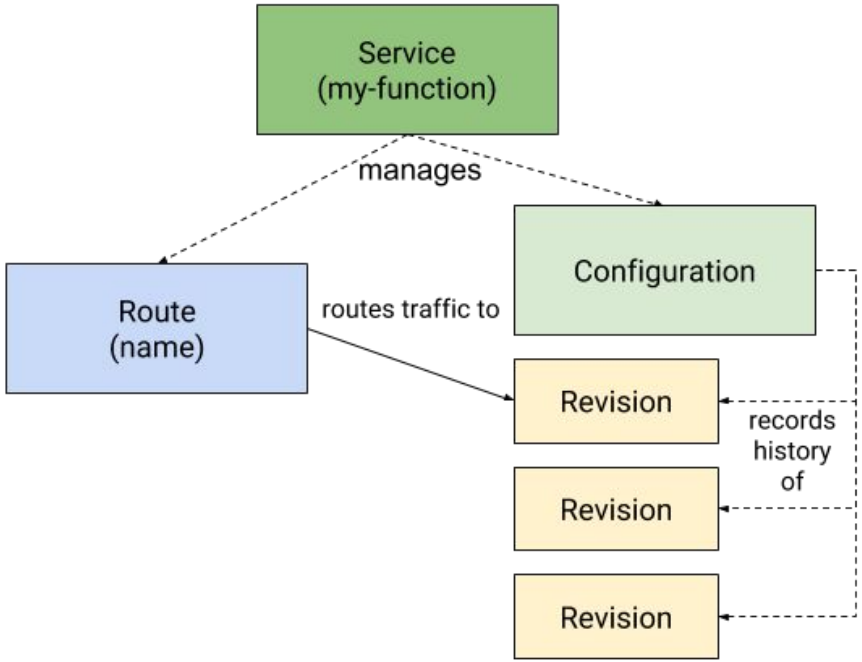
the case for Knative

- **what is Knative?** set of Kube components to facilitate
  - running serverless workloads (as images)
  - manage serverless workloads
  - build serverless workloads => now Tekton Pipeline
- **architecture**
  - collection of Kube CRDs (custom resource definitions)
  - `Service` -> `Configuration` -> `Revision` -> `Route`
  - `Event Source` -> `Broker` -> `Trigger`
- **current state**
  - monthly release cadence — now at version 0.14
  - good pace of contributions from various companies

IBM **Cloud**

@maximilien

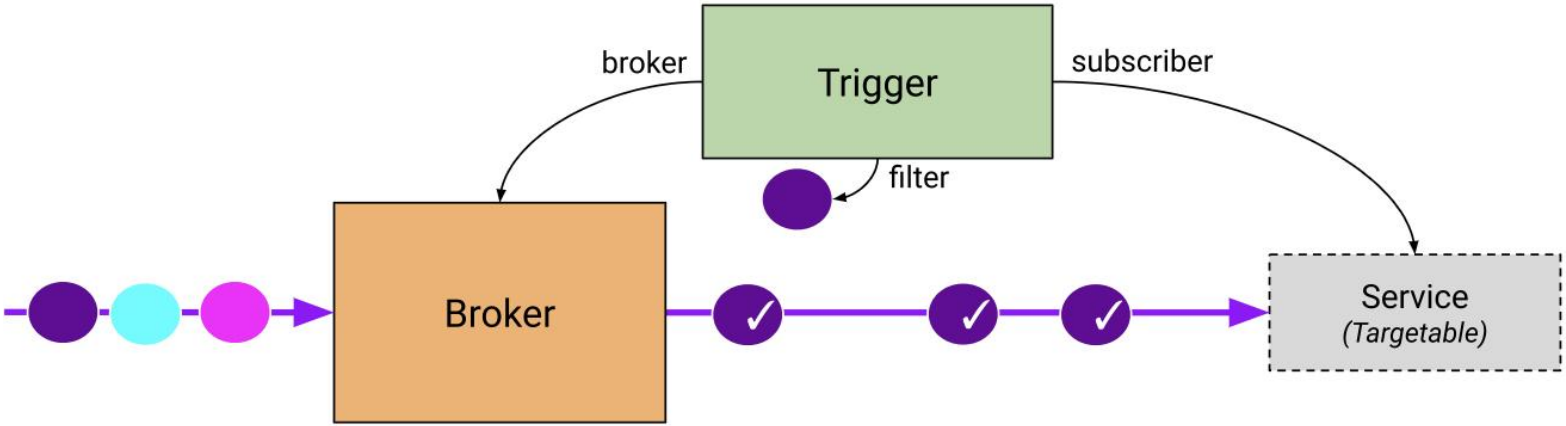
knative overview



IBM Cloud

@maximilien

knative overview (*cont.*)



IBM **Cloud**

@maximilien

community update

- **key repositories**
  - github.com/knative/serving (also includes scaling)
  - github.com/knative/eventing
  - github.com/knative/eventing-contrib
  - github.com/knative/client
  - github.com/knative/client-contrib
  - github.com/knative/pkg and github.com/knative/docs

- **workgroup meetings**
  - open to all, weekly meetings for different groups
  - TOC == technical oversight committee
  - monthly community calls for demo, elections, etc.

IBM **Cloud**

@maximilien

demo

IBM **Cloud**

[@maximilien](https://twitter.com/maximilien)

demo goals

- **micro-services**
  - creating micro-services (micro-functions) that you can test locally
  - creating docker images for your micro-functions
  - running your micro-service as a function

- **serverless**
  - using `kn` to create, delete, and update services or functions
  - using `kn` to connect functions together
  - using `kn` to debug functions

IBM **Cloud**

[@maximilien](https://twitter.com/maximilien)

demo design sketch

- **TwitterFn search function (via Twitter APIs)**
  in: hashtags or string
  in: count - max number of tweets
  out: image URLs from recent tweets

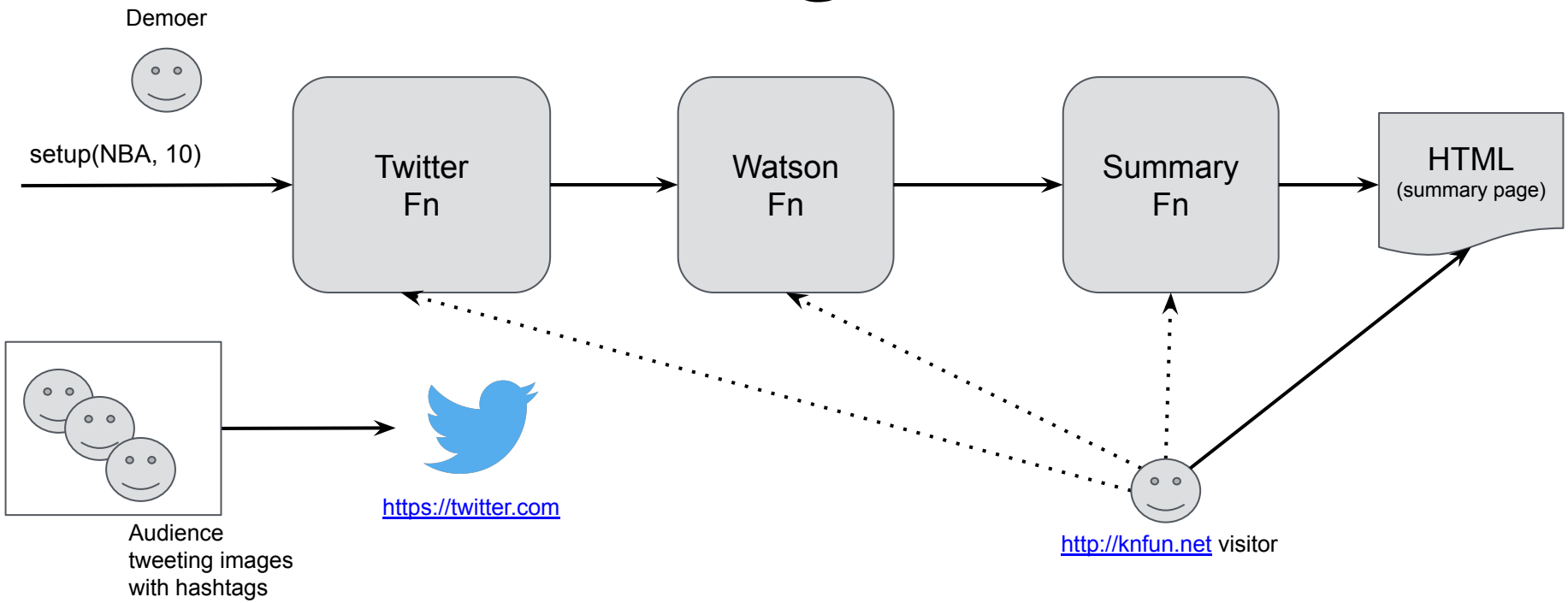
- **WatsonFn image analysis function (via Watson APIs)**
  in: image URLs
  out: image features with probability

- **SummaryFn function**
  - *in:* search string and TwitterFn and WatsonFn
  - *out*: HTML page displaying summary

IBM **Cloud**

@maximilien

demo design sketch



IBM Cloud

@maximilien

demo steps

- **build**

```
$ git clone git:github.com/maximilien/knfun.git
$ cd knfun
$ ./hack/build.sh
```

- **test**

```
$ ./twitter-fn search NBA -c 10 -o json -S -p 8080 --twitter-api-key ...
$ ./watson-fn vr classify http://<image>.jpg -o json -S -p 8081 --watson-api-key
$ ./summary-fn NBA -c 10 -S -p 8082 --twitter-fn-url … --watson-fn-url ...
```

- **image**

```
$ docker build -v -f ./funcs/twitter/Dockerfile -u drmax/twitter-fn
$ docker push drmax/twitter-fn
    ...
```

- **functions**

```
$ ./kn service create twitter-fn --image docker.io/drmax/twitter-fn --env
twitter-api-key ...
...
```

IBM **Cloud**

@maximilien

summary

- **cloud native applications to serverless**
  - cloud native as 12-factor applications
  - serverless takes 12-factor to its natural progression
  - users pay only for what they use!

- **Knative takeaways**
  - builds on popular container orchestrator: Kubernetes
  - provides building blocks for serverless workloads on Kube
    - serve, scale, manage services (functions)
    - source, broker, trigger, and consume events

IBM **Cloud**

@maximilien

summary (*cont.*)

- **IBM's contributions —** ~15 active developers
  - leads for Tekton, eventing, and Knative operator
  - contributing to major components
    - scale to 0, cold-start, vanity URLs, operators
    - kn client (initial), kn plugin, 4 plugins, eventing
    - more coming

- **Knative @ IBM**
  - Knative add-on to the IBM Kubernetes Service (IKS)
  - Knative + Kubernetes as base for our application cloud

IBM **Cloud**

@maximilien

questions?

IBM **Cloud**

@maximilien

@maximilien