Knative Hands-On Workshop:

Why is deploying a Knative service easier than a K8s deployment?

Harald Uebele
@Harald_U

Developer Advocate, IBM

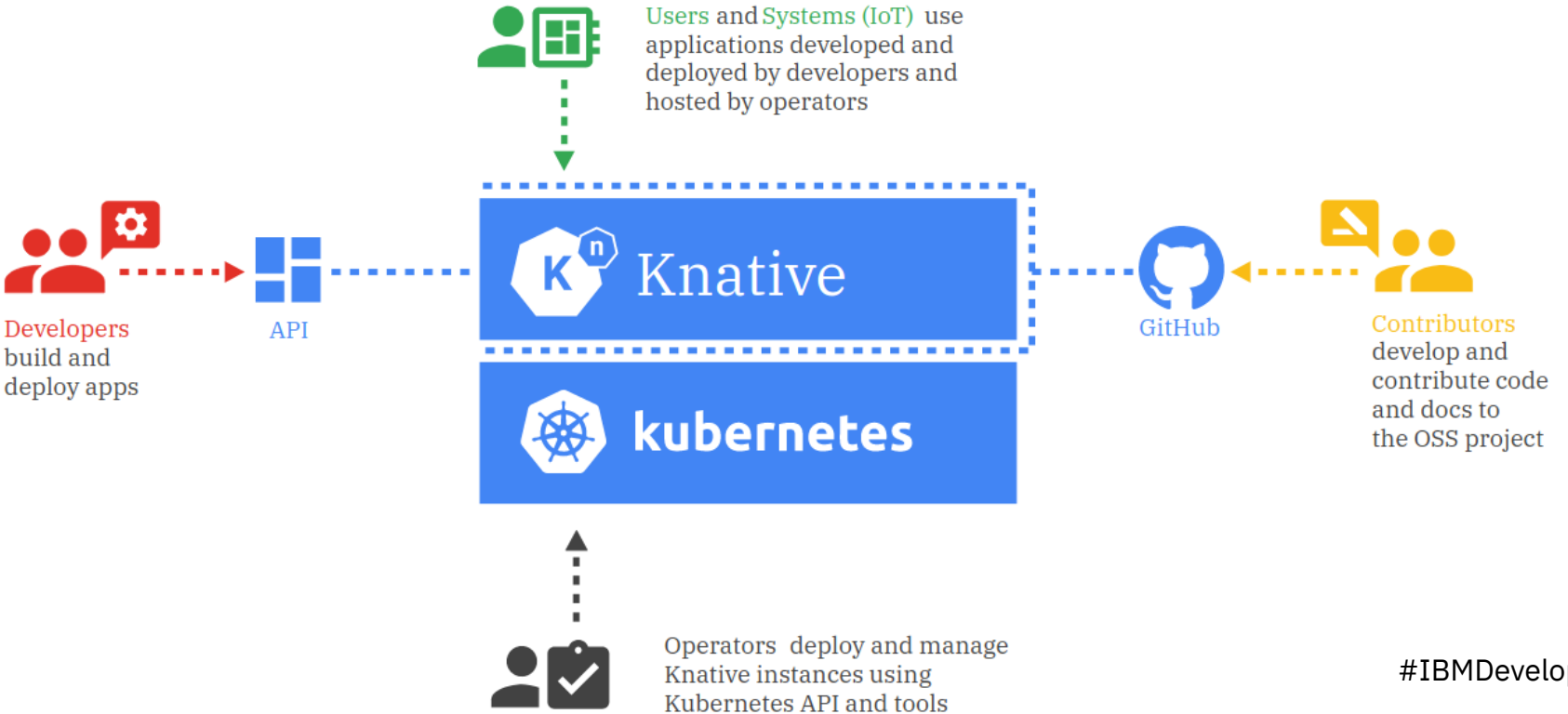
IBM **Developer**

IBM

“Knative solves the *boring but difficult* parts of deploying and managing cloud native services so you don't have to.”

Knative.dev

Knative addresses different personas



#IBMDeveloper

Knative Runtime Contract

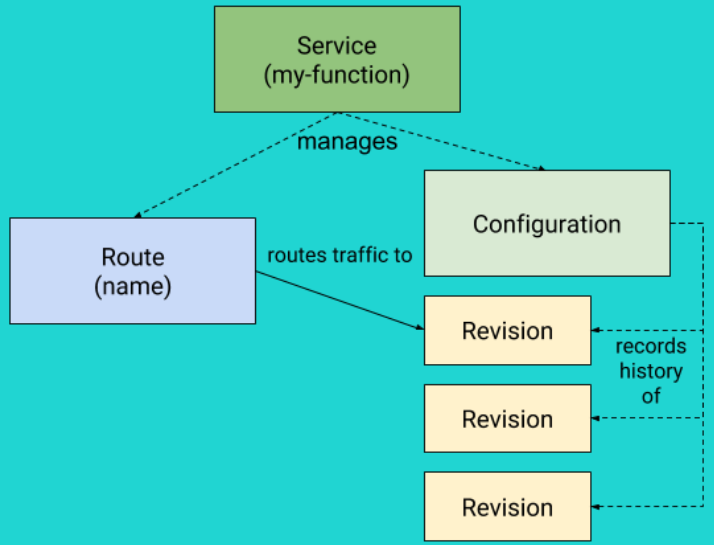
https://github.com/knative/serving/blob/master/docs/runtime-contract.md

The Knative serverless compute infrastructure extends the Open Container Initiative Runtime Specification to describe the functionality and requirements for serverless execution workloads. In contrast to general-purpose containers, ***stateless request-triggered*** *(i.e. on-demand)* ***autoscaled containers*** *have the following properties:*

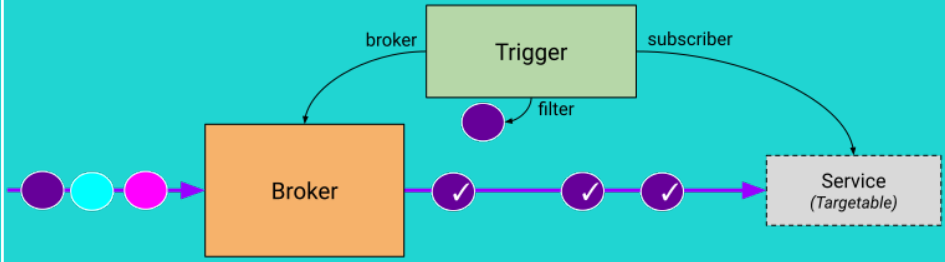
- ***Little or no long-term runtime state*** *(especially in cases where code might be scaled to zero in the absence of request traffic).*
- ***Logging and monitoring aggregation (telemetry)*** *is important for understanding and debugging the system, as containers might be created or deleted at any time in response to autoscaling.*
- ***Multitenancy*** *is highly desirable to allow cost sharing for bursty applications on relatively stable underlying hardware resources.*

Knative Components

Knative Serving



Knative Eventing
(not part of this workshop)



Knative Serving resources

Service

- Manage lifecycle
- Control creation of
  - Route
  - Configuration
  - Revision(s)

Route

- Map network endpoint to one or more revisions
- Traffic Management

Configuration

- Desired state of deployment
- Seperation of code and configuration

Revision

- Snapshot of code and configuration
- Immutable
- Scale up and down

Knative
(service.yaml)

vs.

Kubernetes
(deployment.yaml)

service.yaml:

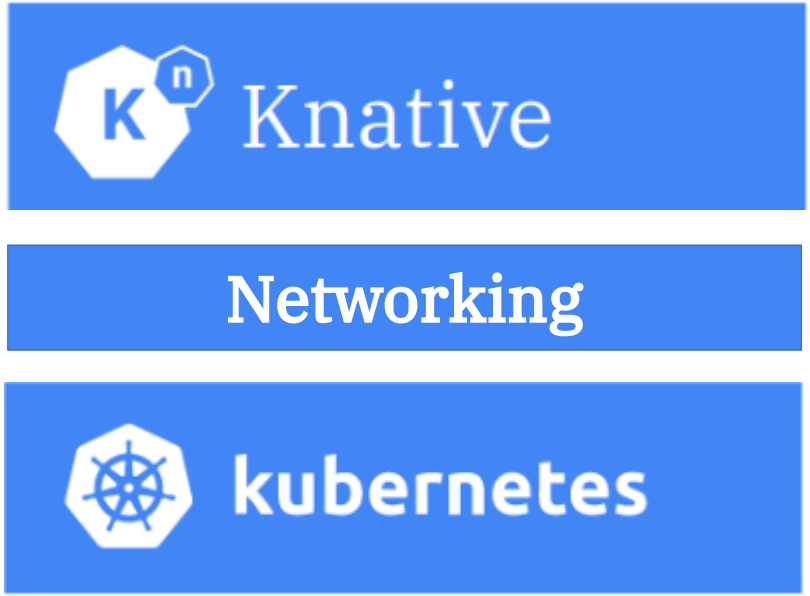
```yaml
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: authors
spec:
  template:
    metadata:
      name: authors-v1
    spec:
      containers:
      - image: docker.io/haraldu/authors:1
        env:
        - name: DATABASE
          value: 'local'
        - name: CLOUDANT_URL
          value: ''
```

deployment.yaml:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: authors
  labels:
    app: authors
    version: v1
spec:
  selector:
    matchLabels:
      app: authors
      version: v1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: authors
        version: v1
    spec:
      containers:
      - image: authors:1
        name: authors
        env:
        - name: DATABASE
          value: 'local'
        - name: CLOUDANT_URL
          value: ''
        ports:
        - containerPort: 3000
          name: authors
---
apiVersion: v1
kind: Service
metadata:
  name: authors
  labels:
    app: authors
spec:
  type: NodePort
  ports:
    - port: 3000
      protocol: TCP
      name: http
  selector:
    app: authors
---
```

Knative Networking Layer

Provides Ingress, Sidecar, etc
Select from:
- Istio
- Kourier (ex 3Scale, now Knative)
- Ambassador
- Contour
- Gloo
- Kong



Question: Is Knative = Serverless?

Answer: It depends!

Yes	• Auto-scaling, scale-to-zero • Only pay for the resources you use • Simplify deployment of code
Maybe	• No features to deal with latency and startup times • Starts container images • Cannot use pre-warmed containers and inject code
No	• If your applications have long startup times and you can't change that (classic Java?) • Either accept latency or prevent scale-to-zero