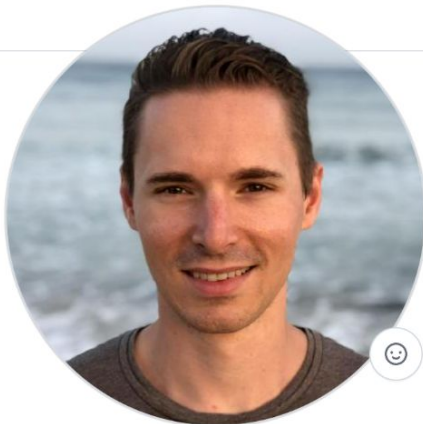# Knative - A story about Serverless Development

Open Source @ Siemens, 23.05.2023, Reto Lehmann

Who am I?

Reto Lehmann
ReToCode

Software Engineer at Red Hat, working on **@knative**, containers, PaaS (OpenShift/Kubernetes) and cloud native stuff in general.

Overview   Repositories 85   Projects   Packages   Stars 171

ReToCode / README.md

Welcome to my github page 👋

I am Reto, a passionate technology enthusiast. I am a fan of continuous improvement and making things easier. I'm a continuous learner and like to see things in a bigger perspective and context. I'm happy when I can move things forward.

🧑‍💻 I'm currently working on Knative at Red Hat.

Pinned
Customize your pins

knative/**serving**   Public

Kubernetes-based, scale-to-zero, request-driven compute

● Go   ☆ 5k   ⑂ 1k

openshift-knative/**serverless-operator**   Public

Main source repository for Openshift Serverless

● Go   ☆ 39   ⑂ 59

raspberry-k8s   Public

oscp/**openshift-monitoring**   Public

Roadmap 🛣️

- Introduction to Knative
- Demo time
- Peek behind the curtains

Knative Overview

Knative is an Open-Source Enterprise-level solution to build Serverless and Event Driven Applications

Serverless Containers in Kubernetes environments.

Try the tutorial	Install Knative on your cluster

Trusted by

IBM Red Hat Google vmware TRIGGERMESH CLOUD NATIVE INTEGRATION PLATFORM

Knative Building Blocks

Knative

Kubernetes-based platform to build, deploy, and manage modern serverless workloads

477 followers · https://knative.dev/ · @KnativeProject · admins@knative.dev · Verified

🏠 Overview   📖 Repositories 31   📋 Projects 19   📦 Packages   👥 Teams 37   👤 People 132

Pinned

serving Public

Kubernetes-based, scale-to-zero, request-driven compute

🔵 Go   ⭐ 5k   🍴 1k

eventing Public

Open source specification and implementation of Knative event binding and delivery

🔵 Go   ⭐ 1.3k   🍴 518

client Public

Knative developer experience, docs, reference Knative CLI implementation

🔵 Go   ⭐ 312   🍴 237
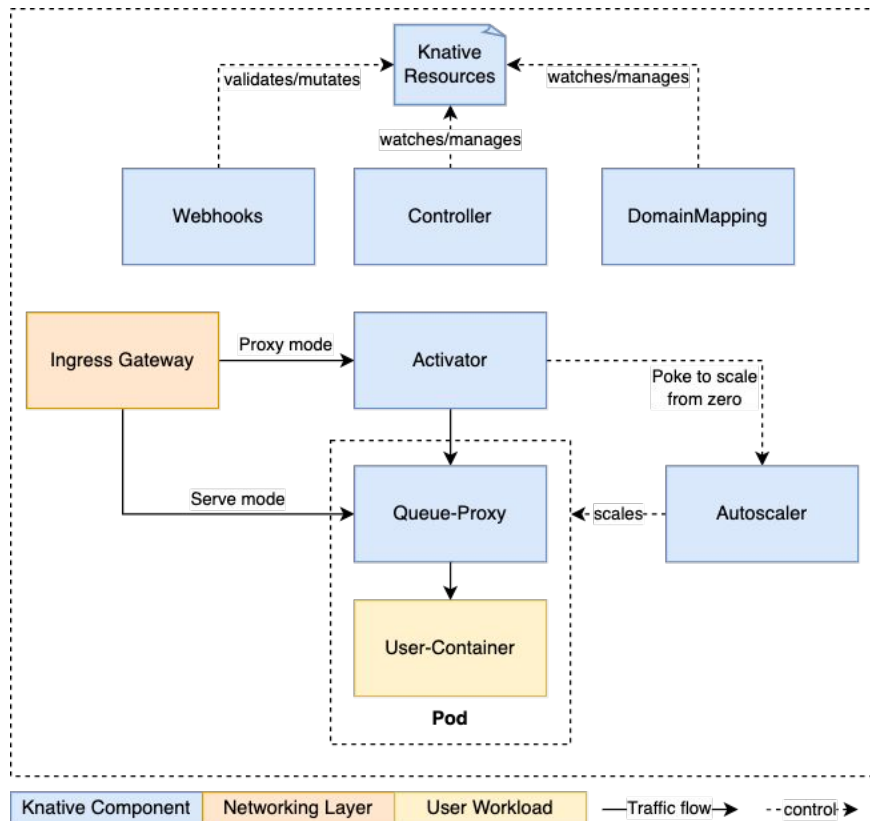
func Public

Knative Functions client API and CLI
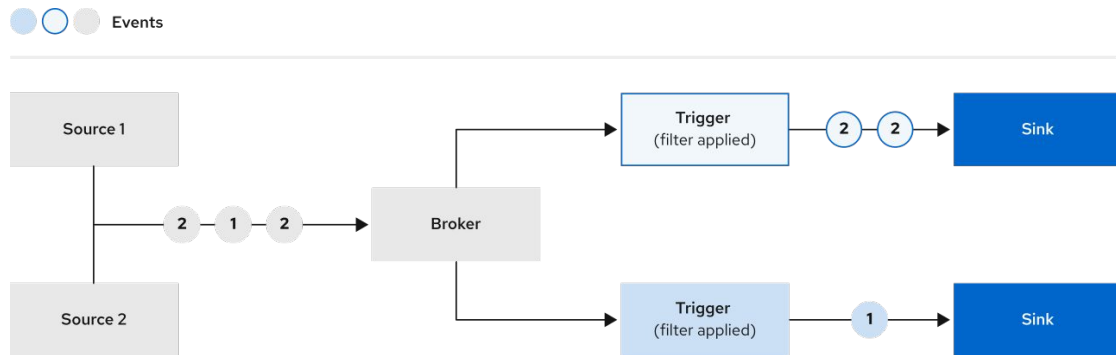
🔵 Go   ⭐ 173   🍴 99

Knative Serving

- Autoscaling
- Request Queuing
- Metrics
- Networking stuff
  - A/B, canary deployments
  - Gradual rollouts
  - Auto TLS
  - Domain mapping

Knative Eventing

- Eventing Abstraction
- Payload as CloudEvents
- Multiple Event Sources
- Logging + Metrics

Knative Client + Functions

- CLI Tooling
- The `kn` CLI allows to manage Knative resources from CLI
- Knative Functions
  - Enables **development** and **deployment** of Functions
  - Provides the `func` CLI
  - Bootstrap functions for multiple languages
    - Node.js
    - Python
    - Go
    - Quarkus
    - Rust
    - Spring Boot
    - TypeScript

```
❯ func
func is the command line interface for managing Knative Function resources

        Create a new Node.js function in the current directory:
        func create --language node myfunction

        Deploy the function using Docker hub to host the image:
        func deploy --registry docker.io/alice

Learn more about Functions:  https://knative.dev/docs/functions/
Learn more about Knative at: https://knative.dev

Primary Commands:
  create       Create a function
  describe     Describe a function
  deploy       Deploy a function
  delete       Undeploy a function
  list         List deployed functions

Development Commands:
  run          Run the function locally
  invoke       Invoke a local or remote function
  build        Build a function container
```

Demo 💻

⌛ Way more stuff, like

- A/B, Canary Deployments, Gradual rollouts
- Tag based routing
- Configuration and Revisions
- Metrics and Observability
- Knative Eventing
- Please refer to the [documentation](#)

Behind the curtains

Open Source Development
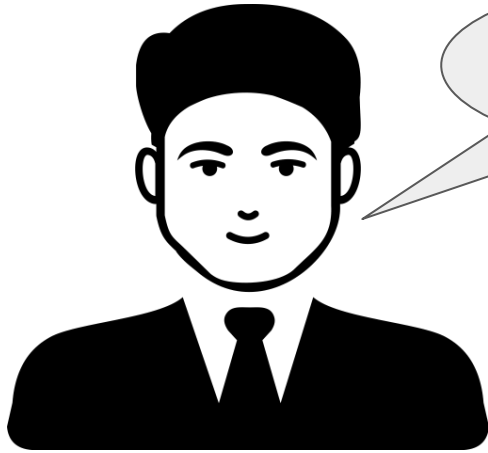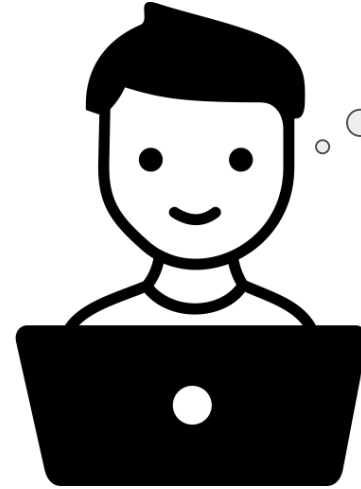
Open Source Development

2. Git clone

```
$ git clone git@github.com:retocode/serving.git

Cloning into 'serving'...
remote: Enumerating objects: 138535, done.
remote: Counting objects: 100% (138535/138535), done.
remote: Compressing objects: 100% (37071/37071), done.
remote: Total 138535 (delta 94630), reused 138190 (delta 94500),
pack-reused 0
Receiving objects: 100% (138535/138535), 83.23 MiB | 12.89 MiB/s, done.
Resolving deltas: 100% (94630/94630), done.
```
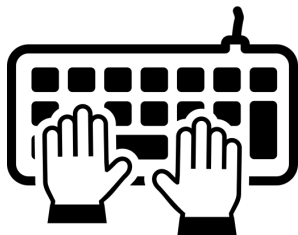
1. Fork

4. Git commit, Git push

3. Hack, hack, hack

```
$ git add .; git commit -m "add new configuration variable";
git push

On branch main
Your branch is up to date with 'origin/main'.
```

The reality is a bit more complicated

Knative

Overview Repositories 31 Projects 19 Packages Teams 37 People 132

Find a repository... Type ▾ Language ▾ Sort ▾

serving Public

Kubernetes-based, scale-to-zero, request-driven compute

app networking serverless function container autoscaler developer-productivity

● Go ⚖ Apache-2.0 ⑂ 1,028 ☆ 4,974 ⊙ 167 (15 issues need help) ⇅ 15 Updated 29 minutes ago

Knative Sandbox

Overview Repositories 50 Projects 6 Packages Teams 71 People 123

Find a repository... Type ▾ Language ▾ Sort ▾

eventing-kafka-broker Public

Alternate Kafka Broker implementation.

● Go ⚖ Apache-2.0 ⑂ 78 ☆ 99 ⊙ 61 (5 issues need help) ⇅ 20 Updated 29 minutes ago

The reality is a bit more complicated

- Massive amount of code
- Domain Knowledge
  - Example 1
  - Example 2
- Shared Components and Packages
- Reasoning in past PRs
- Outdated documentation
- A **LOT** of bash scripts

```
$ tree serving
├── AUTHORS
├── CODE-OF-CONDUCT.md
├── CONTRIBUTING.md
├── DEVELOPMENT.md
├── LICENSE
├── Makefile
├── OWNERS
├── OWNERS_ALIASES
├── README.md
│
…
        └── yaml
SECURITY_CONTACTS
          ├──
code-of-conduct.md
          ├── fields.go
          ├── yaml.go
          └──
yaml_go110.go

2207 directories, 7335
files
```

Planning is chaotic at times

- Not a single team/person/organization responsible
- Multiple tools/processes
- People are no longer working on Knative

knative / **serving** Public

<> Code   ⊙ **Issues** 167   ⇅ Pull requests 15

Filters ▾   🔍 is:issue is:open

⊙ **167 Open**   ✓ 4,197 Closed

Support GRPC and HTTP2 without explicit port labeling

⊙ Open   **rgregg** opened this issue on Jun 6, 2019 · 24 comments

dprotaso commented on Jun 2, 2021   Member ···

@gh-user Are you continuing work on the queue proxy/activator issues?

☺

ZhiminXiang commented on Jun 2, 2021   Member ···

I don't think @gh-user will continue working on it.

☺

Roles (I)

Role	Responsibilities	Requirements	Privileges	Scope
Member	Regular active contributor in the community	Has made multiple contributions to the project	Member of the GitHub Knative org Member of the Knative Slack workspace Edit access to team drive	GitHub Organization
Approver	Approve contributions from other members	Highly experienced and active reviewer and contributor to an area	Entry in one or more OWNERS files in GitHub, and write permissions on one or more repos allowing PRs to be merged	GitHub Directory

More detailed on: https://github.com/knative/community/blob/main/ROLES.md

Roles (II)

Role	Responsibilities	Requirements	Privileges	Scope
Working Group Technical Lead	Set priorities for a functional area and approve proposals Triage incoming issues set milestones, repo labels Roadmap alignment with top-level backlog Mentor new contributors, project members, and approvers Responsible for technical health of their functional area	Sponsored by the technical oversight committee as documented here	Write permissions on one or more repos allowing issues to be manipulated	Working Group

More detailed on: https://github.com/knative/community/blob/main/ROLES.md

Working Groups

- Serving
- Eventing
- Client
- Docs and User Experience
- Functions
- Productivity
- Operations
- Security

Regular public meetings (recorded)

Public agenda and notes

More detailed on:
https://github.com/knative/community/blob/main/working-groups/WORKING-GROUPS.md

Knative Serving Specification

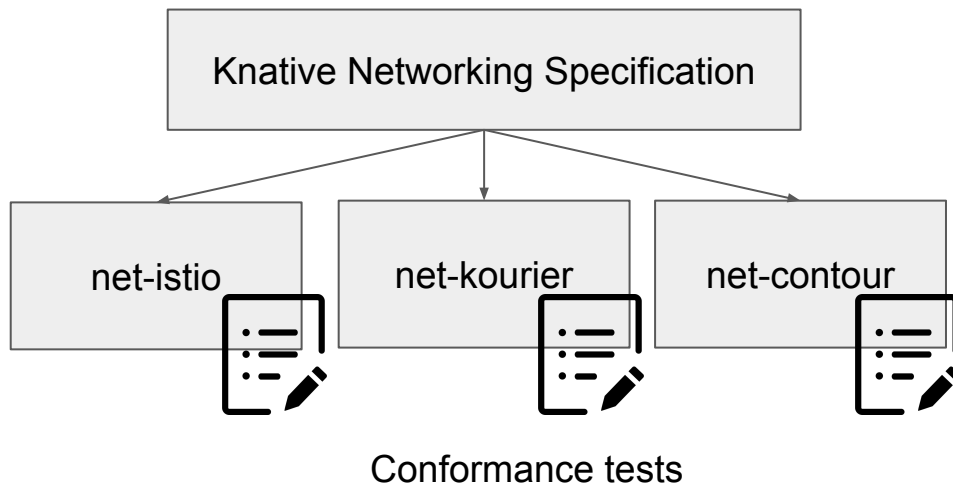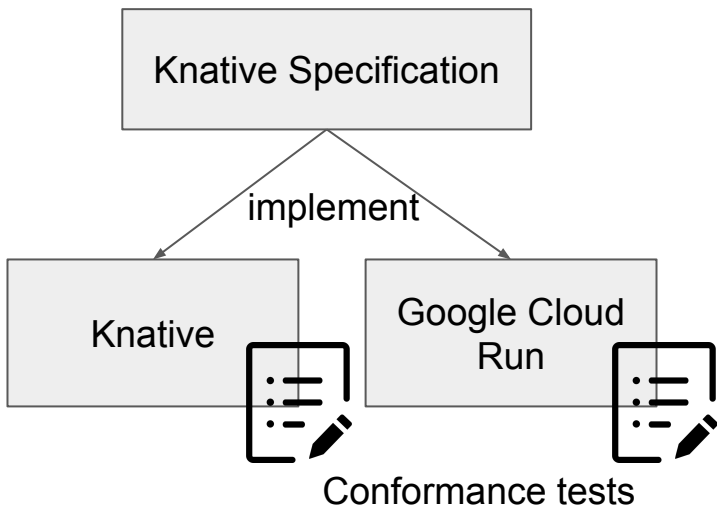The Knative Serving specifications have been moved. See the links below for document locations:

- Knative Serving Conformance Tests

Docs in this directory:

- Motivation and goals - non-normative
- Resource type overview - non-normative
- Knative Serving API Specification
- Knative Serving Runtime Specification

Conformance tests

Conformance tests

CI/CD

- Standardized go-build, go-test, security check, style-check, pr-verification
- Serving
  - ~3000 Unit Tests
  - E2E tests on Kind on GitHub Actions with a matrix of
    - Multiple Kubernetes versions
    - All networking layers
    - Non TLS/TLS/Service Mesh
  - E2E tests on "real clusters" in GKE with the same matrix
- Cross-Repository tests (imports go tests)

GitHub Actions

prow

59 checks passed

✓ 🔘 build
✓ 🔘 build / Build
✓ 🔘 style / changes
✓ 🔘 analyze / Analyze CodeQL
✓ 🔘 test / Unit Tests
✓ 🔘 verify / Verify Deps and Codegen
✓ 🔘 test (v1.24.7, kourier, runtime)
✓ 🔘 test (v1.24.7, kourier, api)

Behind the Red Hat curtains

Creating a product out of OSS

- OpenShift Serverless is Knative
- Still Open Source: https://github.com/openshift-knative
- Changes are always done **upstream** if possible
- Extended operator to support installation on OpenShift
- A lot of additional testing for OpenShift

openshift-knative

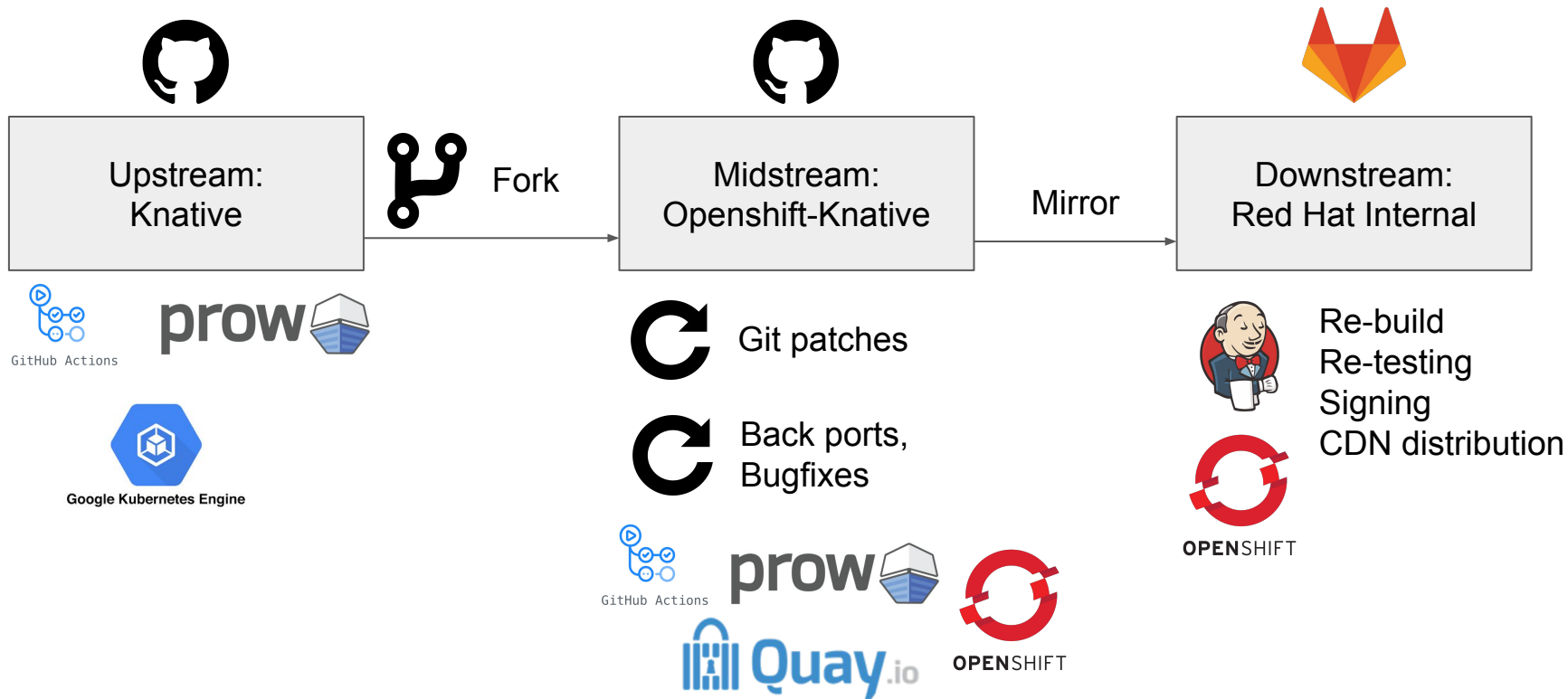5 followers   https://www.openshift.com/learn/to...   serverless-support@redhat.com

The pipeline

Personal summary on OSS development

- Can be chaotic at times, but very welcoming, helpful and open community
- It can be hard to get started, look for guidance (like good-first-issues) and mentorship
- Good documentation and communication is key
- Having a good coverage of features/functionality by tests enable automation
- Bash is *still* a pretty common ground
- Git patches are a pretty solid way to extend/adapt some OSS code that you do not have full control over (you can also do pretty hacky stuff, but that is better discussed over 🍻😆)

Thank you