Exotic Runtimes

Ruby and Wasm on Kubernetes and GitOps Delivery Pipelines

Kingdon Barrett

Weaveworks

2023-05-09

Lightning Talk

Gotta go pretty fast

- Try not to talk so fast
- Don’t want to lose you
- Many twists and turns
- Rabbit-shocker timer to help keep honest! (See Flux and Flagger logos?)

Intro

- Hi
  I'm Kingdon Barrett
- Find me on YouTube or Mastodon
- youtube.com/@yebyen
- hachyderm.io/@yebyen

Job

- Weaveworks: Developer Experience
- OSS Engineer @ WW since 2021
- Second S is for ~~Smooth Operator~~ (no, it's Open Source **Support**!)
- Work on Flux (Slack/ web/ community maintainer, former Flux v1 maintainer)

Flux

- Flux Bug Scrub - weekly fluxcd.io/#calendar
- What: OS **Support** Engineer
- (I try to use our OSS deeply)
- Lean into fully OSS solutions

Flux Talks

bit.ly/gitopscon2023

Intro (me)

- On YouTube - I'm new here
- Let's Study: Arabic
- Cloud Jockey: `%radio` DJ
- Live Coding: Ruby + Kubernetes
- (Please mash like & subscribe button)

Wasm and Ruby

- **What are we here for today**
- What is “untrusted code”
- Why do we want to run it
- Healthy skepticism about (yes, even our own) code

Ruby

- Can Wasm? run it
- Yes
- Why would we do that?
  - This is a serious question
  - Do you know why Wasm? (What is Wasm?)

Why Wasm

- Secure Foundation
- “Bytecode Alliance”
- Portable artifacts
  - with a degree of language independence

Why Wasm

- Frankly I cannot sell Wasm
  - No commission either
- If you take it with you, I get nothing
  - I think it will be useful
  - Let’s find out together

Why Kubernetes

- For Flux and GitOps
- If you chose Kubernetes,
  you already know why you did (!)
- Declarative, versioned, immutable
  artifacts describe a desired state
- Self-healing infrastructure

Compiled Languages

- Rust
- Go
- JavaScript, TypeScript
- C#
- ... (value for you all as well)

History as a Rubyist

- I used Ruby since 2002(?)
  - Thanks Eivind - Enlightenment, E16, E17, `#gah` on ~~EFnet~~, ~~Freenode~~, IRC (I'm old now)
  - First "permanent job"
    at Metrix Matrix in Rochester, NY
  - Second "permanent job"
    at University of Notre Dame OIT, South Bend

Why Ruby?

- I know it better than other languages
- Comfort and familiarity
  - Top Notch Debugging ++
  - Bundler, Fibers, Ruby 3.0
- MVP: for faster time to market

Ruby Solutions

- To run a website
- To connect a database
  - scrape content from internet
  - To build an IRC bot
  - No compiler, duck typing, object orientation, imperative, monkey patching, ease of use

More Ideas: Ruby Solutions

- To build a K8s Operator, of course because why not!

Web Assembly in Ruby

- Ruby: interpreted language
- gem: wasmer-ruby
- gem: wasmtime-rb
- Run Wasms in Ruby
- What is a Web Assembly?

Runtime Format

- Can run Ruby in Wasm?
- Yes, but first...
- Wasm is a binary format
- Wasm also builds libraries
  - Include it in other programs

Ruby in Ruby?

- Consider not doing this
- No theoretical benefit afaict
- It was the first thing I tried
- I could not make it work
- Let’s try the other thing

Web Assembly

- Call functions from it
- Ship memory around
- Export functions to it
- Use a compiler, or...
- System Interface (WASI)

Features: Format

- What is a system interface?
- Stdio, filesystem, (restricted) HTTP/S
- There is no network
- How do you run a server?
  - WAGI (like CGI!) offload responsibility of conn

Omitted Features

- Has no string type, tough limitations
- Numbers and well-defined data structures only(ish) - hard for Ruby
- Allocate memory, make ptr
- Pass ptr to str+length/size

Ruby and pointers

- I don’t want to do pointer math at all
- Could not figure out how do:
  - Wasm as library
- I spent some time on this, couldn’t make it work in Ruby unfortunately

Ruby and Wasm lib

- I need string params and return values
- Reverted to WASI for this
- We can parse the output, pass in fs dir
- Now let’s try to solve a real problem with some Wasm in Ruby!

What is Spin?

- My first entrypoint to Wasm
- Great docs about Ruby capabilities, references to relevant projects + docs
- “Serverless” - compose Wasms + run
- Test locally, no Kubernetes needed, no “linking” - works like controller/router

What is Spin?

- “Serverless” framework
- Test locally
- Run on Fermyon Cloud

What is Spin?

- “Serverless” framework
- Test locally
- (Run on Hippo Factory)
  - This is OSS Summit!

What is Spin?

- “Serverless” framework
- Test locally
- Run on Fermyon Cloud
- (Run on Hippo Factory)
  - ~~This is OSS Summit~~!

What is Spin?

- Serverless framework
- Test locally
- (Run on Hippo Factory)
  - This is GitOpsCon!

What is Spin?

- Serverless framework
- Test locally
- (Run on Kubernetes!)
  - This is GitOpsCon, after all

Why are we here?

- Hope to gain:
  - Testability
  - Reusability
  - Type safety between languages
- Capacity for polyglot teams to work together, benefit from specialization in many languages

How about we dive in?

- I built some things in Wasm
  - Break misconceptions
  - Follow good examples
- How are we going to use it?
- Let’s solve a real problem now

Problem to explore

- GitHub Packages problem - DX needs to know how many downloads over time
- fluxcd/flagger/
  - pkgs/container/flagger
  - GitHub does not expose this value on any API afaict, so we scrape some HTML and parse it!

I built some things

- EKS cluster: Find on GitHub [kingdonb/eks-cluster]
  - with Flux bootstrap (eksctl+flux 2.0.0-rc.2)

I built some things

- Blog service: GitHub kingdonb/taking-bartholo
- GitOps via Helm Controller
- Helm + Helmet library chart
  - At this point I understood pain of running Wasm+Kubernetes (not pain from Ruby... yet)

I built some things

- I began to understand some things
  - Fermyon isn't using K8s or Helm
  - This would be v. hard without Flux (used Flux OCI to ship content separately from runtime, a novel application of Flux's OCI Artifacts!)

I built more things

- Kubernetes operator: GitHub [kingdonb/stats-tracker-ghcr]
  - Fetch from URL (in Ruby)
  - Write to file, pass in fs context
  - Parse HTML (in rust)
  - Return number as string (WASI!)

Finally

- Kubernetes operator: GitHub kingdonb/stats-tracker-ghcr
  - Parse number (back in Ruby)
  - Store the number we parsed out of scraped content in CRD status
  - (Come back and retrieve it later)

Based on

- Kubernetes operator: (GitLab) [tobiaskuntzsch/kubernetes-operator]
  - Wonderful example to build with Ruby
  - Register CRD, Register `upsert`
  - Reg. `delete` - manages Finalizers

Based on (dependency)

- Kubeclient gem: GitHub [ManageIQ/kubeclient]
  - Also easy to use
  - Server-side apply only (!)
  - (about SSA, Flux uses this too)
    - can account for admission controllers, wait for ready, … lots of benefits here!

Out of time

- Lightning talk - that's all for today!
- Dive into topics further
  - OSS Summit later this week
    - More Ruby: ContainerCon (Wed 11:00am)
    - Go/TypeScript: OpenGovCon (Thu 4:05pm)

      with co-presenter Will C, Defense Unicorns

Operator isn't finished

- Let's do it live (for real, I mean today)
- On YouTube (.com/@yebyen)
- We will hear how it went tomorrow, and more about each project
- (Don't worry, 98% finished already `:D`)

Let's do it live

- I wrote this app, so far, in an hour (live)
- Code is there, a few things not working yet
- Only missing a bit... we'll tour the code!
- First Rust app - be gentle (it was fun!)

Thank You

- Visit us for the rest of the week: bit.ly/gitopscon2023
- Find these slides: kingdonb/cdcongitopscon2023-slides on GitHub

Thank You

github.com/kingdonb/cdcongitopscon2023-slides