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!

Intro

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

Job

- Weaveworks: Dev Experience
- OSS Engineer since 2021
- Second S is for ~~Smooth Operator~~ (no, it's Support!)
- I work on Flux (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 + K8s
- Plz mash like & subscribe

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?

Why Wasm

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

Why Wasm

- Frankly I cannot sell Wasm
- No commission either
- If you take it, 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
- Self-healing infrastructure

Compiled Languages

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

Why Ruby

- I used Ruby since 2002(?)
  - Thanks Eivind (attyz)
- Comfort and familiarity
  - Top Notch Debugging ++
  - Bundler, Fibers, Ruby 3.0
- 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 needed, duck typing, object orientation

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
- There is no network
- How do you run a server?
  - WAGI (it's like CGI!)

Omitted Features

- Wasm has no string type
- Numbers and well-defined data structures only(ish)
- 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 figure unfortunately

Ruby and Wasm lib

- I need string return values
- Reverted to WASI
- We can parse the output
- Now let's try to solve a real problem

What is Spin?

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

What is Spin?

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

What is Spin?

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

Why are we here?

- Hope to gain:
- Testability
- Reusability
- Type safety btw languages
- Capacity for polyglot teams

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 real problem now

Problem to explore

- GitHub Packages problem - DX Engineer solution
- fluxcd/flagger/
  - pkgs/container/flagger
  - Need to know how many downloads for each package

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 enabled via Helm Controller
- Helm + Helmet library chart
  - At this point I began to understand

I built some things

- At this point I began to understand some things
  - Fermyon isn’t using K8s or Helm
  - This would be hard without Flux (used Flux OCI to ship content separately from runtime, novel!)

I built more things

- Kubernetes operator: 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: 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 ex. 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 - all for today!
- Dive into topics further
  - OSS Summit later this week
    - Ruby: ContainerCon (Wed 11:00am)
    - Go: OpenGovCon (Thu 4:05pm)

    with co-presenter Will Christensen

Operator isn't finished

- Let's do it live (today)
- On YouTube (.com/@yebyen)
- (Hear how it went tmrw.)
- It is 98% finished already :D

Let's do it live

- I wrote this in an hour live
- Code is there
- Only missing a tiny bit
- My first Rust program!

Thank You