Kubernetes on Edge:

Bringing Your Code to Constrained Places

Dejan Bosnac

Software Engineer
Red Hat

Steven Wong

Software Engineer
VMware

Kilton Hopkins

Chief Technical Officer
Edgeworx

Thursday October 13, 2021 - 2:30pm Pacific
https://sched.co/lV6q

Agenda

What is WebAssembly? Where could you use it?

Demo:

- Pros and cons of using traditional containers versus upcoming technologies such as WebAssembly and WASI.

- Mechanisms of deploying built artifacts to the actual hardware at Edge.

- The evolving state of running WebAssembly workloads using Kubernetes.

How to participate in the Kubernetes IoT Edge Working Group.

WebAssembly

What Is It?

A W3C open standard for implementing portable programs using byte code that at least in theory could run on any hardware and any underlying OS or platform.

Where could it run?

Initially it was used to run client side code, in web browsers. More recently it is being used on the server side, and in edge applications unrelated to a browser. It is not tied to any particular hardware type or OS

edge   mobile   gateway   cloud

How can you implement a web assembly program?

All popular languages have at least some level of support for transformation to wasm executable form.

How good is the performance and efficiency?

Implementations usually use ahead-of-time or just-in-time compilation to run on a host at "near native" speed. Interpreters are also possible and are slower.

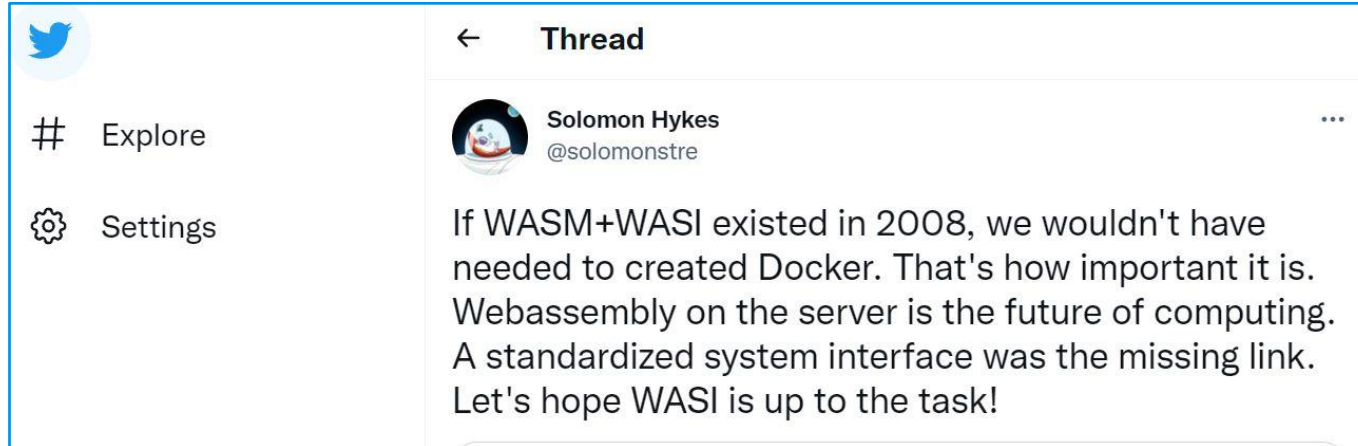# How does WebAssembly compare to Docker

Similarities

> **Thread**
>
> **Solomon Hykes**
> @solomonstre
>
> If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

- Both define a transportable form.

- Both implement a form of "sandboxing for security.

Differences

- Docker is much more mature, with a bigger supporting "landscape".

- WebAssembly: more portable, smaller binary, less memory overhead, faster startup latency, high security by default. Limited network access. Single threaded. Difficulties with use of some forms of language libraries.

What if there was a way to extend or enhance parts of the Docker landscape to handle WebAssembly?

Demos

In demos, Dejan and Kilton will talk about:

- Real life maturity. Current Challenges
- Building for different architectures
- Accessing peripheral devices from WebAssembly applications
- How to build for arm on a non arm – issue for Docker containers, not an issue for wasm
- Wasm runtime –needs for a WASI environment. Krustlet
- Support for files and sockets access.
- Support for cloud events, serverless.
- Integration with Kubernetes.
- How to build a WASM image
- Security challenges
- Managing
- Runtime requirements
- Device access
- Integrating with existing Edge
- Futures

Dejan Bosanac

Software Engineer, Red Hat

@dejanb

Kilton Hopkins

CTO, Edgeworx

@kiltonhopkins

Laying down the groundwork

Requirements for processing WebAssembly workloads at the edge

- WA Runtime

- Agent

- Edge connectivity handling
  - Why pure k8s environment may be unsuitable
  - Edge native technologies allow scale

- Repository of binary code (.wasm) files

Two types of edge deployment

No edge device interfacing

- Latency benefits

- Avoiding bandwidth constraints

- Enhanced security

- Privacy

Processing data from edge devices

- Need WASI or similar

- Host environment becomes application specific

- Reminiscent of the Android HIDL/AIDL structure for hardware abstraction

Diving into the WebAssembly edge future

WASI and the current state of the art

- Core system calls being handled first (files, network, etc.)

- Principles of portability and security: proper focus but slower build-out

Edge-specific challenges

- WebAssembly modules are the new microservices

- It's still going to be a multi-microservice world

- Exchange of data between WebAssembly modules

- Dynamic access to underlying devices and resources

Integrating WebAssembly with existing edge

- Easiest path is integration with existing edge technologies

- WebAssembly runtime choice drives available features at this time

- Side-by-side operation of WA and containers seems likely

- Use cases will drive specific constrained device WA advancements

WebAsssemby Resources
Where to learn more

- Dejan's demo: https://github.com/dejanb/drogue-dht-py

- Wasm on Kubernetes
  - CloudEvents WASI Demo: https://github.com/dejanb/ce-wasi-example
  - Krustlet Kubernetes Kubelet in Rust for running WASM: https://github.com/krustlet/krustlet
  - Wasmtime "small and efficient runtime for WebAssembly & WASI": https://wasmtime.dev
  - Wasm-to-oci: https://github.com/engineerd/wasm-to-oci
  - Hippo WebAssembly Paas: https://deislabs.io/posts/introducing-hippo/
  - CNCF sandbox project https://wasmcloud.dev/

- General WebAssembly Links
  - Blog post introducing WASI: https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/
  - Popular WA runtime Wasmer: https://wasmer.io/
  - WebAssembly: The Future of Cloud Native Distributed Computing

The Kubernetes IoT Edge Working Group

How to get involved and learn more

Link to join the group 👫👤 ✉

- https://groups.google.com/forum/#!forum/kubernetes-wg-iot-edge

Regular Working Group meetings:

USA WG Meeting Wednesday 9am PT, every 4 weeks, next on November 3

APAC WG meeting Wednesday 5am UTC every 4 weeks, next on October 19

- See Kubernetes contributor calendar for details. Agenda+Notes link

Link to join Slack

- https://kubernetes.slack.com/messages/wg-iot-edge

Video Recordings

- https://www.youtube.com/playlist?list=PL69nYSiGNLP0rfR9tvfcfd52_W1GUr3eA

White paper
Edge security white paper

https://github.com/kubernetes/community/tree/master/wg-iot-edge

Speaker contact info

Dejan Bosanac
Red Hat
@dejanb

Steven Wong
VMware
@cantbewong

Kilton Hopkins
Edgeworx
@kiltonhopkins

Q&A

IoT edge WG Slack

https://kubernetes.slack.com/messages/wg-iot-edge

Thank You

This deck is available here:
https://sched.co/lV6q