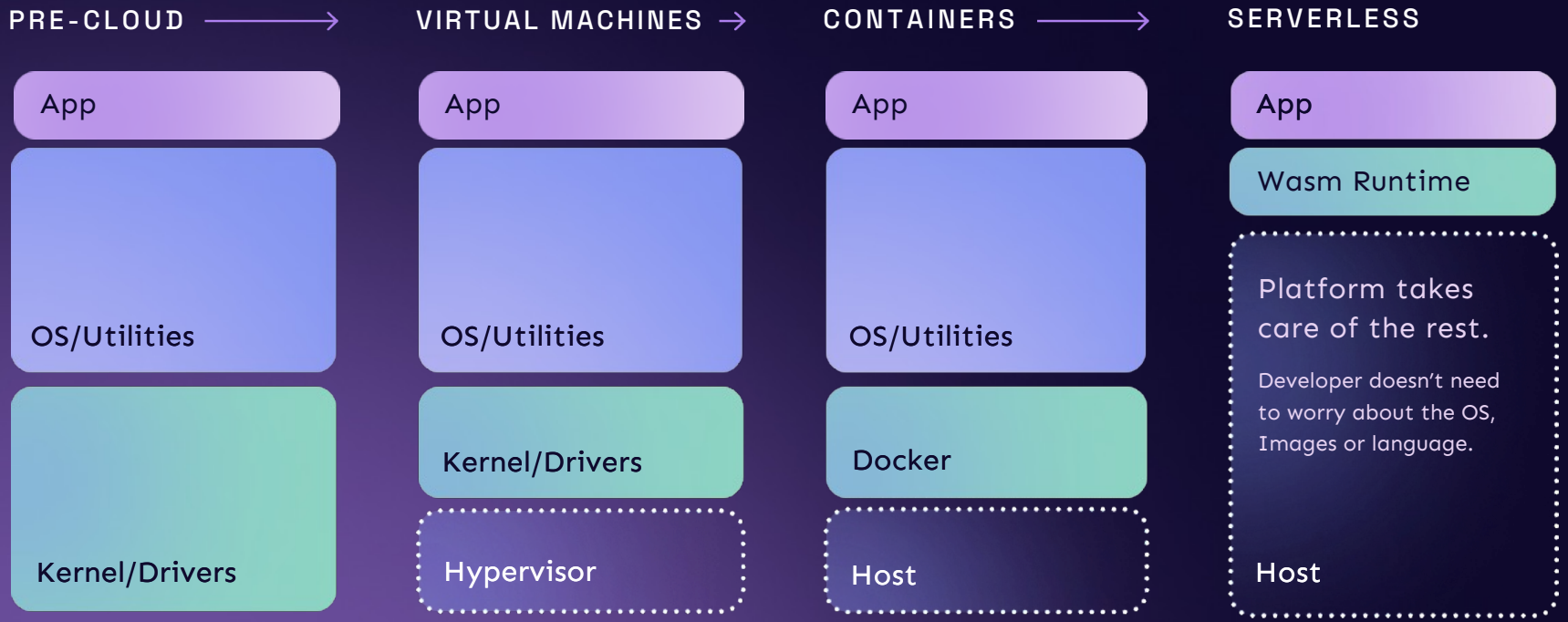
FERMYON

The Future of the Cloud is WebAssembly

Sohan Maheshwar

Lead Developer Advocate @ Fermyon

Cloud computing evolution

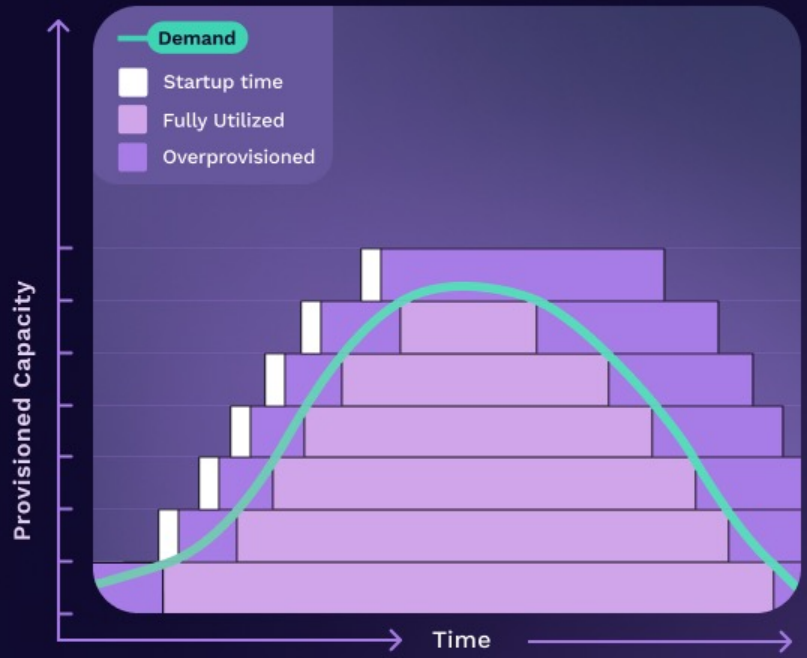


PAIN POINT

Containers are too expensive, over-consuming resources.

[A]cross 50 of the top public software companies currently utilizing cloud infrastructure, an estimated **$100B of market value is being lost** among them due to cloud impact on margins [...]

Source: https://a16z.com/the-cost-of-cloud-a-trillion-dollar-paradox/



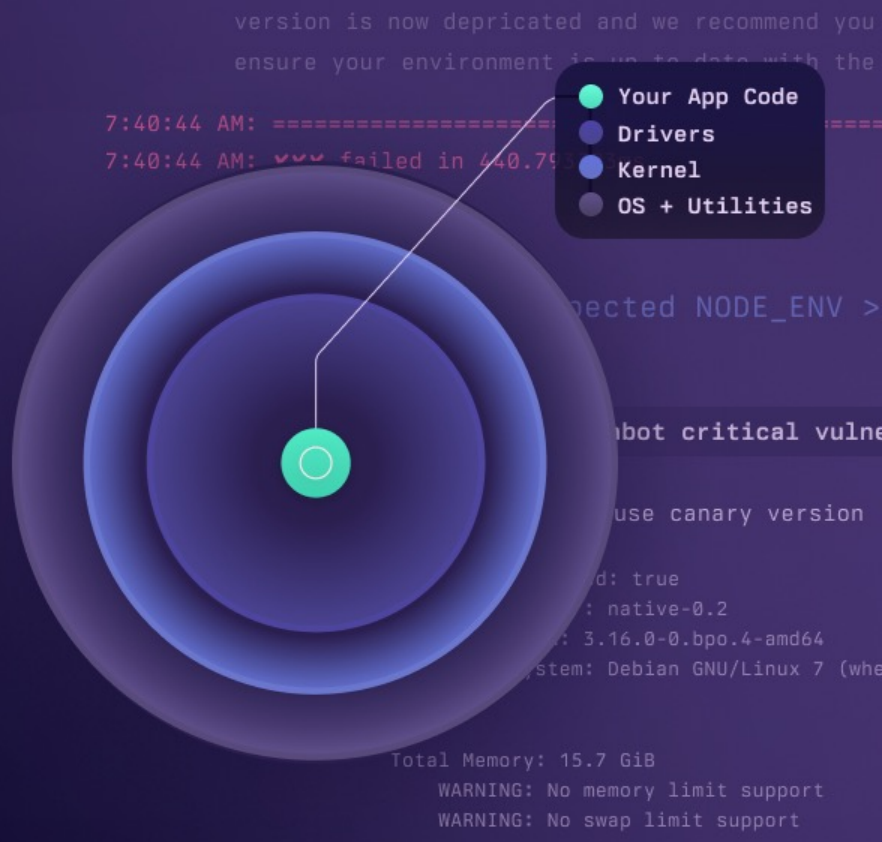
FERMYON

PAIN POINT

Apps configs are complex, with too many layers of operational dependencies.

Modern apps comprise frameworks, language dependencies and libraries that need to ship alongside your code in the cloud, adding extra compatibility and stability overhead.

Portability and simplicity are often sacrificed - locking your app to the tools and architecture of the day.

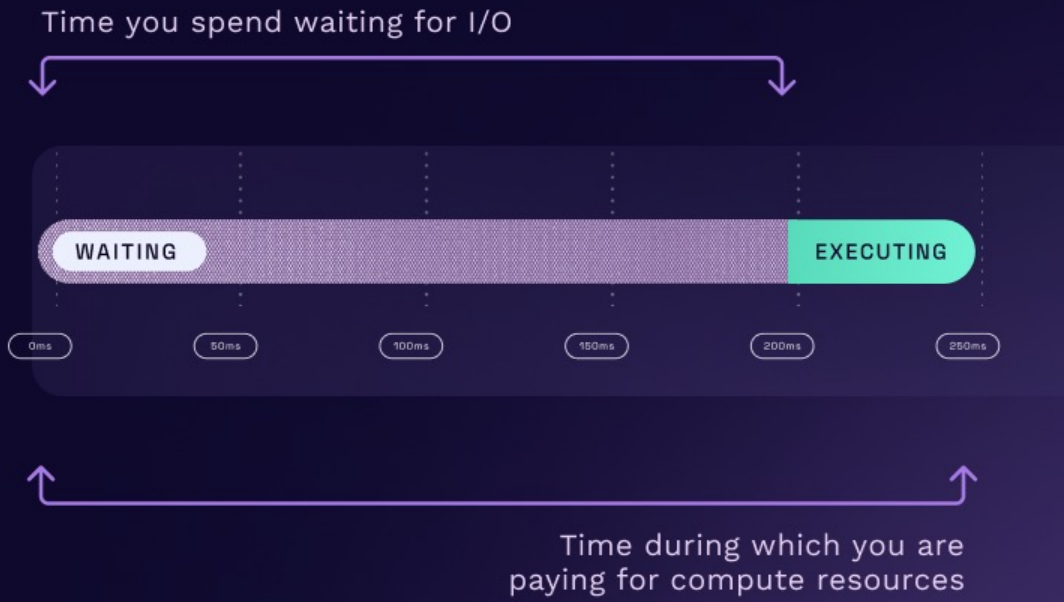


FERMYON

PAIN POINT

Serverless has a Cold Start problem.

Solutions like AWS Lambda frequently take 2-3 seconds to wake up and start executing. Often the startup delay is orders of magnitude longer than the execution time itself, requiring complex and expensive workarounds.



FERMYON

The next wave of cloud compute will be powered by WebAssembly

FERMYON

Solomon Hykes / @shykes@hachyderm.io

@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!

FERMYON

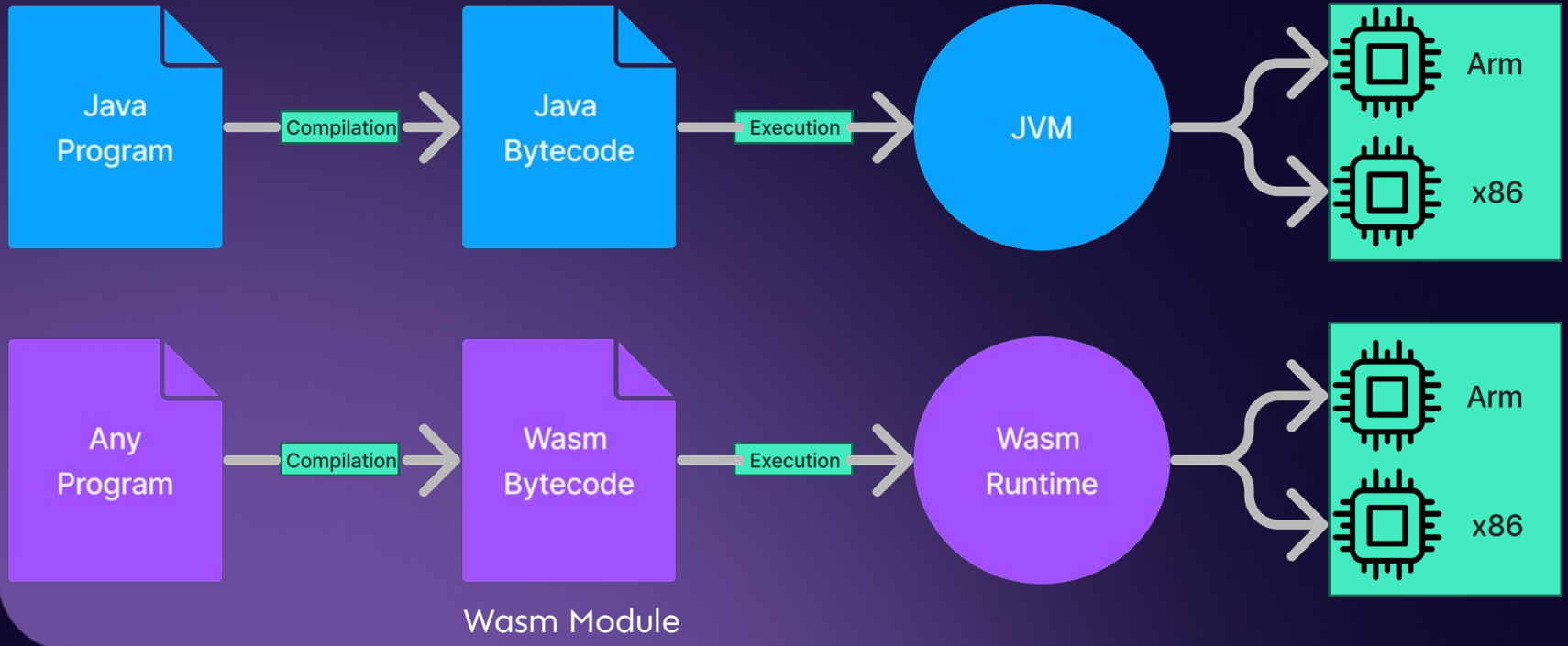
What Is WebAssembly?

The boring answer: It's just another bytecode format

A few things to know about WebAssembly

- Wasm is just another name for it
- Designed as a portable compilation target

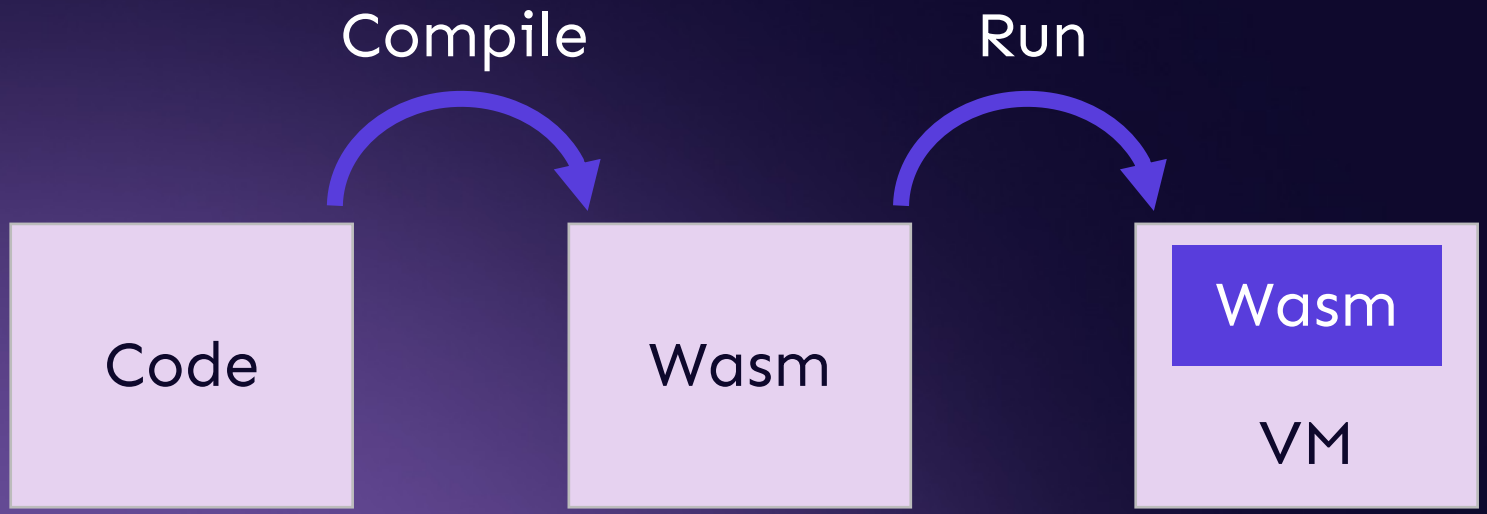
Wasm is another bytecode format



A few things to know about WebAssembly

- Wasm is just another name for it
- Designed as a portable compilation target
- Originates from the browser, now also available outside
- "compile once" and then run that code on any number of targets

Compile and Run



WASI: A new kind of System Interface

TL;DR: it allows you to run WebAssembly outside of the browser

- Access to several operating-system-like features, including files and filesystems, clocks, and random numbers
- Independent of browsers, so it doesn't depend on Web APIs or JS
- It extends Wasm's sandboxing to include I/O.

Compilation and Language Support

WebAssembly Support in Top 20 Languages

This reports on the top 20 languages from RedMonk's ranking. Some languages, like CSS, PowerShell, and "Shell", don't really have a meaningful expression in Wasm. However, we have left them here for completeness.

Language	Core	Browser	WASI	Spin SDK
JavaScript	✅	✅	⏳	✅
Python	✅	⏳	✅	✅
Java	✅	✅	✅	⏳
PHP	✅	✅	✅	❌
CSS	N/A	N/A	N/A	N/A
C# and .NET	✅	✅	✅	✅
C++	✅	✅	✅	❌
TypeScript	✅	⏳	❌	✅
Ruby	✅	✅	✅	❌

https://www.fermyon.com/wasm-languages/webassembly-language-support/

Getting started with Wasm

FERMYON

Introducing Spin

- The open-source tool for building WebAssembly serverless apps
- Create a new serverless with just a few commands.

https://developer.fermyon.com/spin

FERMYON

The framework to compose serverless WebAssembly apps.

OPEN SOURCE

15+ LANGUAGES

4.6K GITHUB ★

SIMPLE CLI

github.com/fermyon/spin

SPIN

Spin is an open source project, built with open standards like WASI, Wagi and the WebAssembly Component Model.

AT A GLANCE:

Serverless AI NEW

Quickly test and run inferencing workloads with LLMs.

Key/Value Store

Easily persist data in your apps with a built-in KV store.

Powerful CLI

Easy to create, run and deploy projects - in as little as 66 seconds.

NoOps SQL Database

Add SQLite data to your app with an always-available SQLlite DB.

COMPOSING APPS:

- HTTP & Redis Triggers
- Relational Database Support
- Variables & Secrets Rotation

DEV EXPERIENCE:

- Supports almost any programming language
- Easy to debug with included helper commands

FERMYON

DEMO

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

*https://00f.net/2023/01/04/webassembly-benchmark-2023/

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

Startup Time

Startup times comparable with natively compiled code

Only 2.3x slower than native*

*https://00f.net/2023/01/04/webassembly-benchmark-2023/

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

Startup Time

Startup times comparable with natively compiled code

Only 2.3x slower than native*

Portability

Build once, run anywhere!

Same build (JIT) works across OS and platform arc

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

Startup Time

Startup times comparable with natively compiled code

Only 2.3x slower than native*

Portability

Build once, run anywhere!

Same build (JIT) works across OS and platform arc

Security

Sandboxed execution

Capability based security model

FERMYON

FERMYON

How this will change cloud computing

Gradually, then suddenly

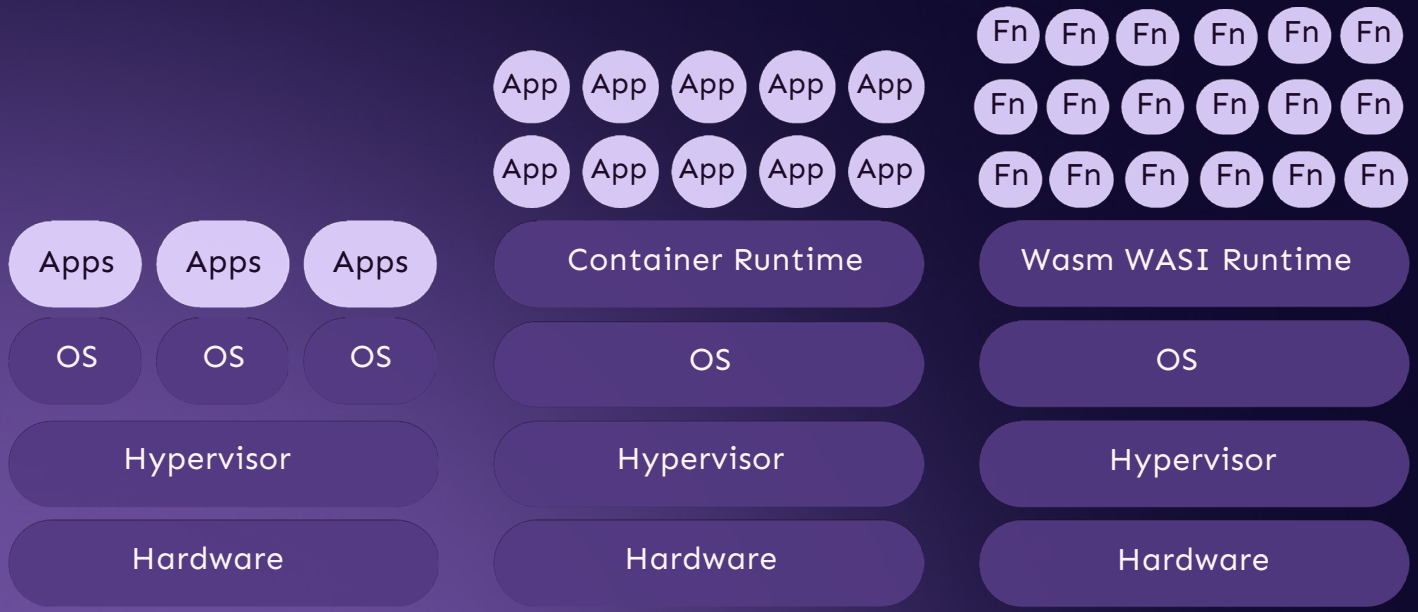
Multitenancy

Multiple independent applications running in a shared environment

- Brings costs closer to value
- Value of a system based on long-term average traffic
- Cost of running a system based on short-term peak traffic

https://brooker.co.za/blog/2023/03/23/economics.html

Multitenancy is Increasing with Each Wave of Cloud Computing



Characteristics of an Ideal Serverless Unit

Isolation	Overhead & Density	Performance	Fast Switching	Soft Allocation	Compatibility
Safe to run multiple functions on the same hardware	Run 1000s of functions on a machine, with minimal waste	Consistent and near-native performance	Quick start and cleanup of functions	Can over-commit resources (CPU, memory, etc.)	Use your favorite libraries and hosts

Firecracker: Lightweight Virtualization for Serverless Applications, 2020, Agache et al https://www.usenix.org/system/files/nsdi20-paper-agache.pdf

Comparing Serverless Units

Characteristic	MicroVM	WebAssembly
Isolation	Sandboxed (via Firecracker VMM + KVM)	Sandboxed (via linear memory and capabilities-based security)
Overhead & Density	1000s per node (48 core, 382 GB RAM, 3360 GB disk)	1000s per node (8 cores, 32 GB RAM, 100 GB disk)
Performance	Near native	Near native
Fast Switching	125 ms (startup to cleanup)	<1 ms (startup to cleanup)
Soft Allocation	Run in production with oversubscription ratios as high as 10x	Untested
Compatibility	Linux + KVM only. Most software is compatible unless specific hardware requirements	OS and platform-agnostic bytecode. Only supports WASI compatible libraries.

Firecracker: Lightweight Virtualization for Serverless Applications, 2020, Agache et al https://www.usenix.org/system/files/nsdi20-paper-agache.pdf

SpinKube

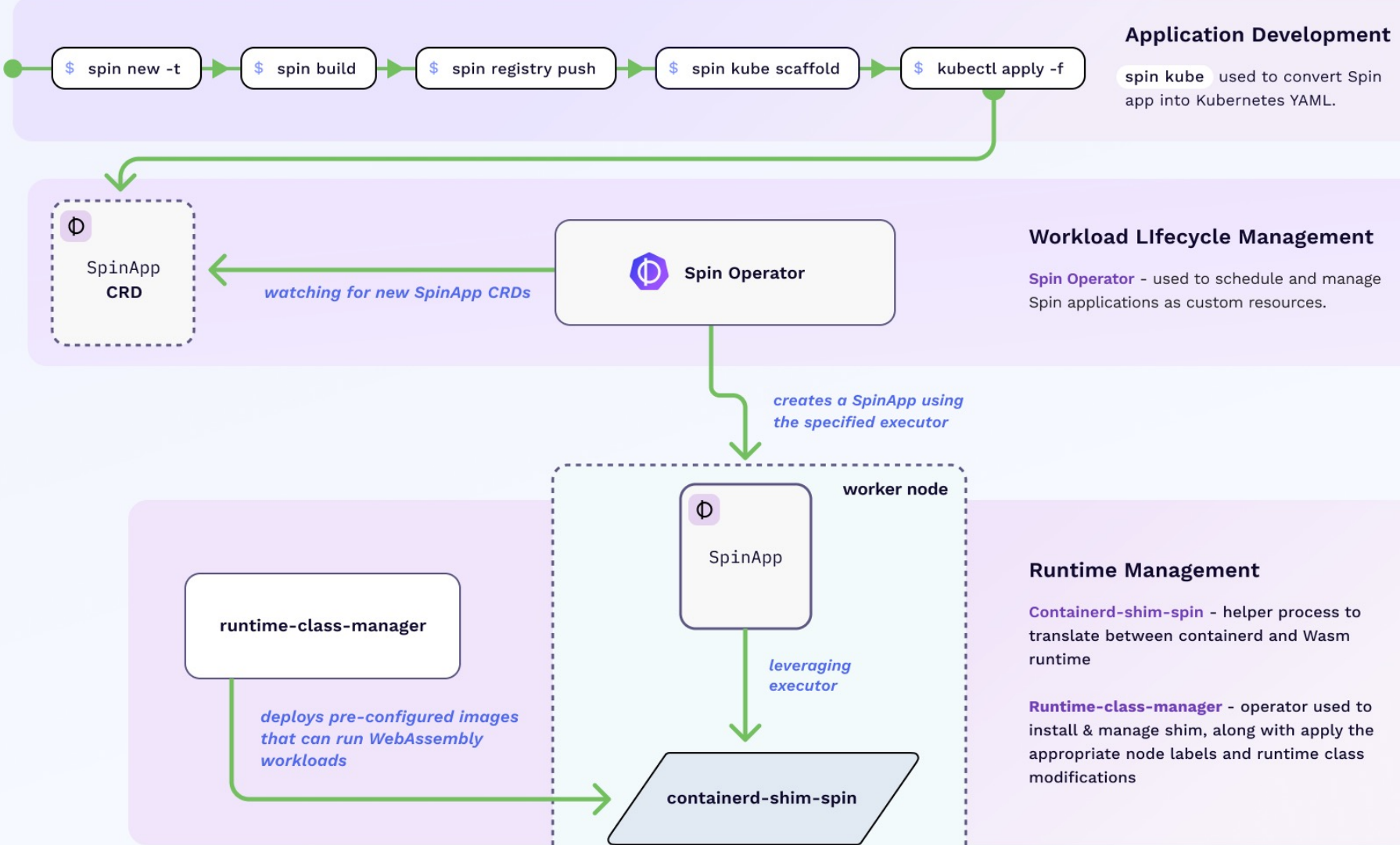
- Hyper-efficient serverless on Kubernetes, powered by WebAssembly
- Open-source project that streamlines developing, deploying and operating WebAssembly workloads in Kubernetes.

Microsoft REPLY LIQUID SUSE FERMYON

SpinKube

- Artifacts are significantly smaller in size compared to container images.
- Artifacts can be quickly fetched over the network and started much faster
- Substantially fewer resources are required during idle times.

www.spinkube.dev


$ spin new -t
$ spin build
$ spin registry push
$ spin kube scaffold
$ kubectl apply -f
Application Development
spin kube used to convert Spin app into Kubernetes YAML.
SpinApp CRD
watching for new SpinApp CRDs
Spin Operator
Workload LIfecycle Management
Spin Operator - used to schedule and manage Spin applications as custom resources.
creates a SpinApp using the specified executor
worker node
SpinApp
runtime-class-manager
leveraging executor
deploys pre-configured images that can run WebAssembly workloads
containerd-shim-spin
Runtime Management
Containerd-shim-spin - helper process to translate between containerd and Wasm runtime
Runtime-class-manager - operator used to install & manage shim, along with apply the appropriate node labels and runtime class modifications

Fermyon Platform for Kubernetes

- 50x workload density - Deliver over 5,000 serverless apps per Kubernetes node.
- Massive reductions in serverless cold start delays (sub 1 ms) and throughput (3-5x)
- Increased capacity and resource efficiency means lower infra spend for your team.
- Highly portable apps and workloads that can be run on a variety of CPU, OS and cloud vendor architectures.

fermyon.com/platform

FERMYON

Platform For Kubernetes

OPEN SOURCE

SPIN

SELF-HOST IN YOUR KUBERNETES

OPEN SOURCE

SpinKube

FERMYON

ENTERPRISE (MORE FEATURES)

Platform For Kubernetes

ANY CLOUD PROVIDER

HOST ON FERMYON

FREE+PAID TIERS

FERMYON

Cloud

CLOUD INFRA, STORAGE, DOMAINS & SERVICES ARE INCLUDED

Next Steps

- Build your first Wasm app using Spin github.com/fermyon/spin
- SpinKube & Fermyon Platform for Kubernetes
  fermyon.com/blog/introducing-spinkube-fermyon-platform-for-k8s
- YouTube: youtube.com/@fermyontech/

Thank You!

linkedin.com/in/sohanmaheshwar/

developer.fermyon.com

Join our Discord server!

Check out Spin!

FERMYON