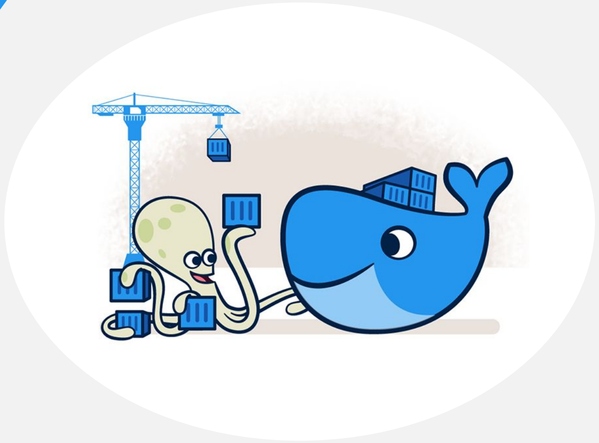
What's New in Docker + Wasm?

WEBASSEMBLY



docker

Speaker

- Developer Advocate at Docker
- Author @ https://collabnix.com
- ex-Docker Captain
- Docker Community Leader, Bangalore
- ARM Ambassador
- Worked @ Dell EMC, VMware, Redis & CGI
-

docker

Community-driven Projects

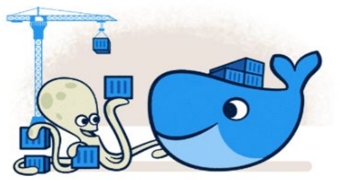
COLLABNIX

List of major projects contributed by the community contributors

https://dockerlabs.collabnix.com

https://kubetools.collabnix.com

https://kubelabs.collabnix.com



Docker Labs

With 4,500 GitHub stars, Dockerlabs is an initiative within the Collabnix community that focuses specifically on Docker-related learning, training, and hands-on labs.

KubeTools

With over 1,100 GitHub stars, Kubetools is a curated list of over 350+ Kubernetes tools and utilities targeted at DevOps Engineers

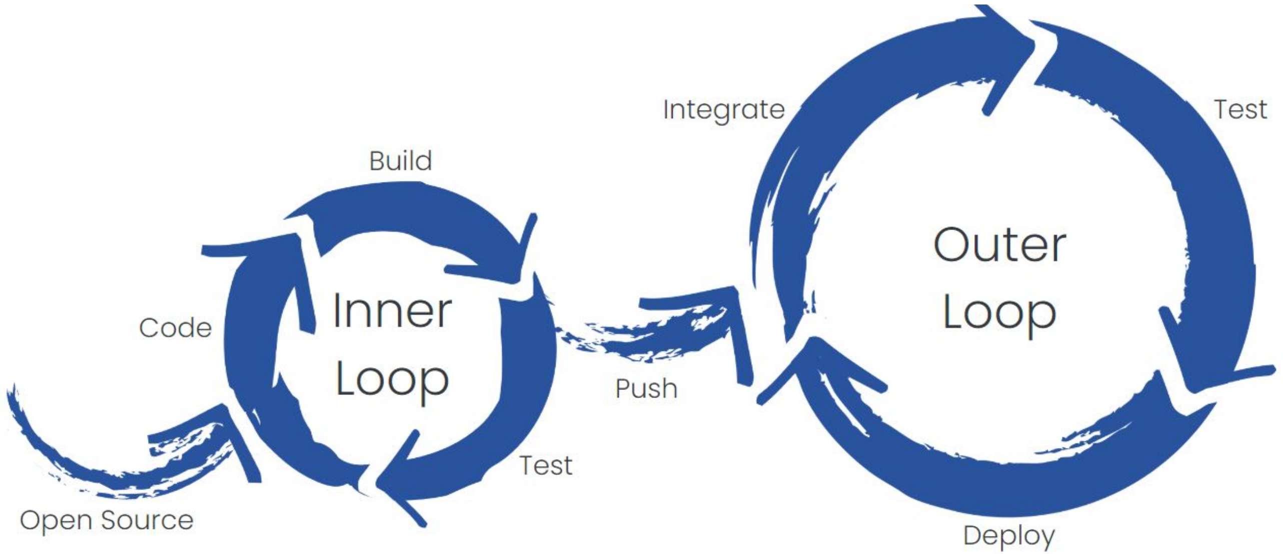
KubeLabs

Kubelabs just crossed 2.2k GitHub stars is an initiative within the Collabnix community that focuses on Kubernetes-related learning, training, and hands-on labs.

Introducing WasmLabs Repository

https://github.com/collabnix/wasmlabs

Docker is Uniquely Focused on Developer Success



Trusted Images
Docker Desktop
Docker Ecosystem
Delivery Platforms

20M+ Active Developers 450+ Trusted Partners

Docker Desktop

Speed

- Docker init
- VirtioFS Support
- Compose File Watch
- Vpnkit ⇒ gVisor

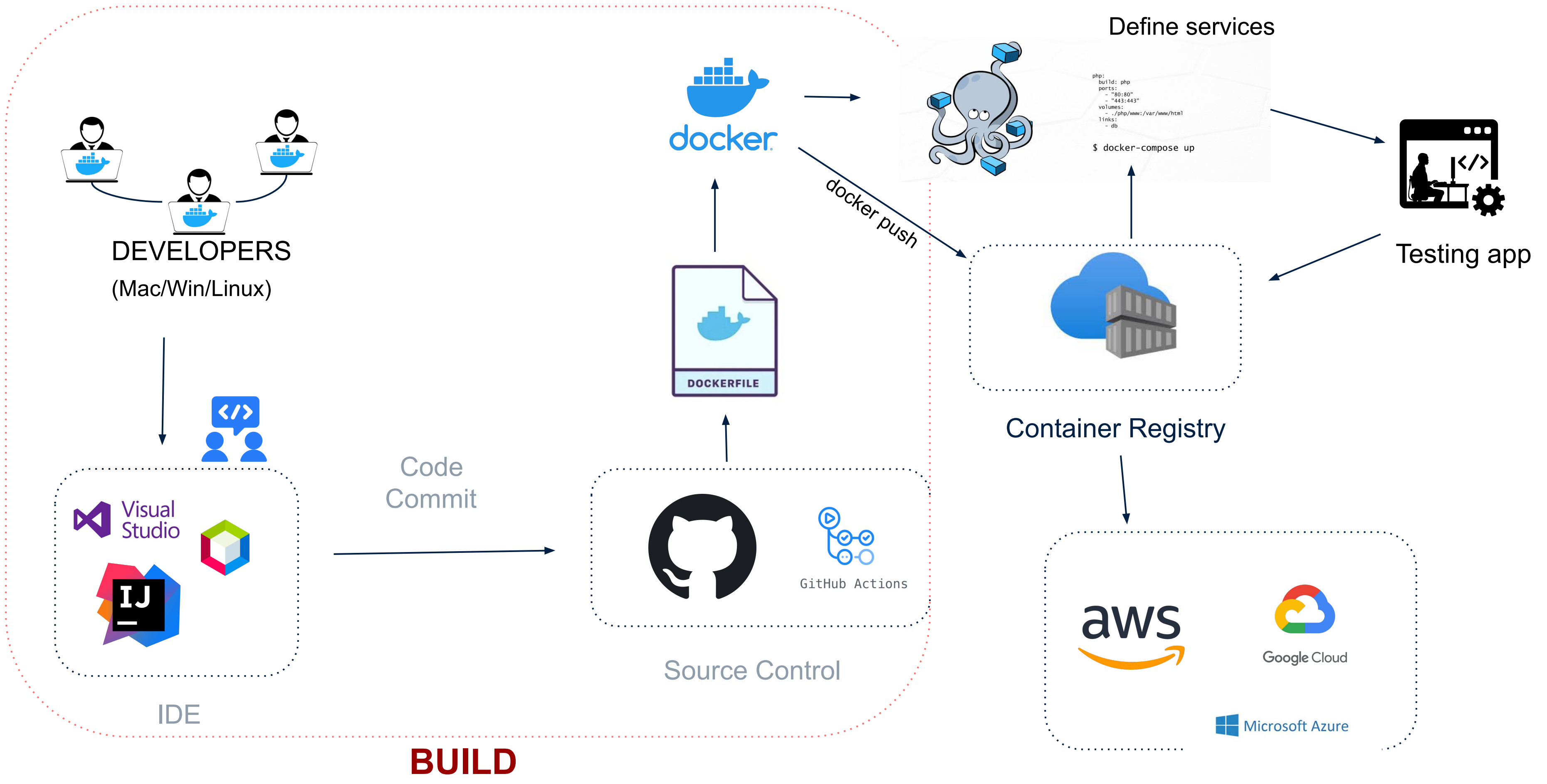
Security

- Docker Scout
- Attestations

Choice

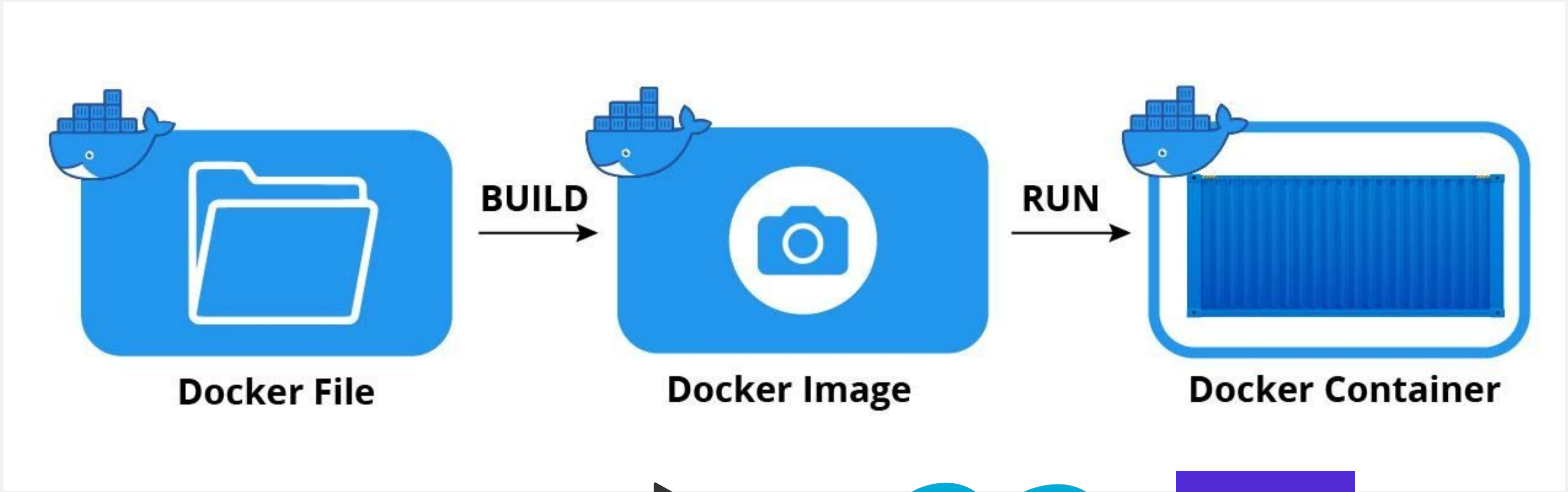
- Docker Extensions
- Docker Sponsored Open Source Projects
- Rosetta 2
- Wasm

Inner-Loop Development Workflow



```
php:
  build: php
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./php/www:/var/www/html
  links:
    - db

$ docker-compose up
```

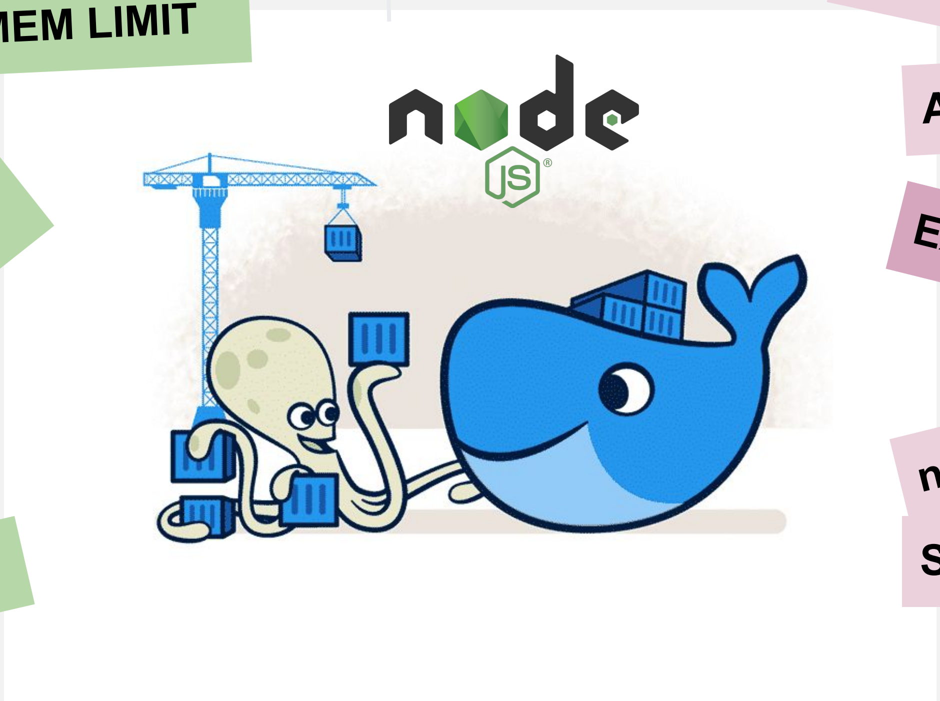
A Typical Docker Workflow



"Do's"

"Don'ts"

- node:<tag>
- USER node
- MEM LIMIT
- HEALTHCHECK
- SCANNING
- COPY
- Multi-Stage
- .dockerignore
- buildx –platform



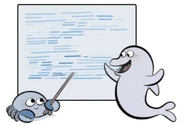
- node:latest
- ADD
- USER root
- EXPOSE db_port
- node_modules
- SIGTERM

Introducing

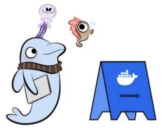
Docker Init

Docker init

Simplified Docker Assets Creation

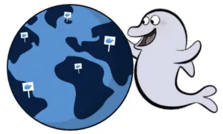


Saves Time and Effort



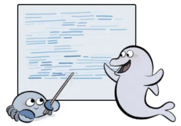
Better Project Organization

Enhanced Portability

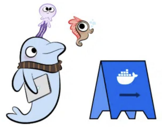


Docker init

Simplified Docker Assets Creation

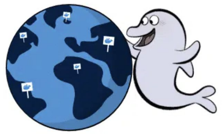


Saves Time and Effort



Better Project Organization

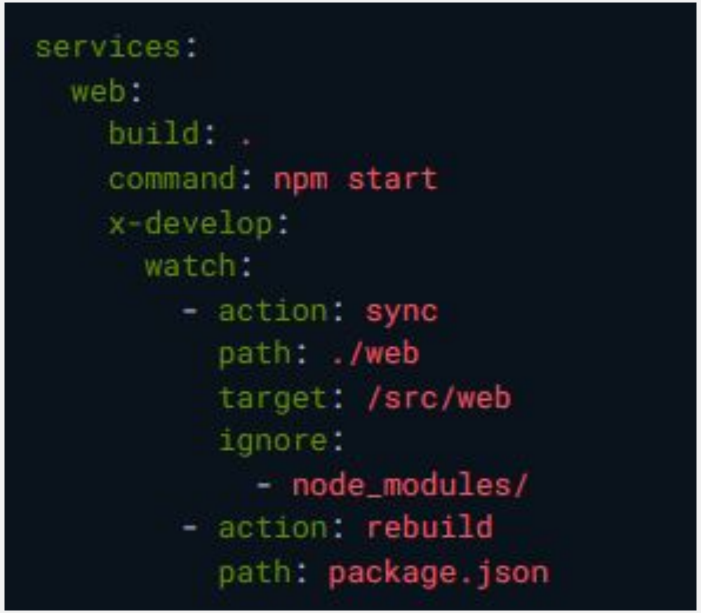
Enhanced Portability



Compose Watch

Compose Watch

- New Experimental Feature
- Automatically updates your compose service containers while you work
- Blazing-fast file synchronization supporting live update



```yaml
services:
  web:
    build: .
    command: npm start
    x-develop:
      watch:
        - action: sync
          path: ./web
          target: /src/web
          ignore:
            - node_modules/
        - action: rebuild
          path: package.json
```

How it works?

- Automatically builds a new image with BuildKit and replaces the running service container
- Add an x-develop section to your services in the compose.yaml file
- Configure it with a list of paths to watch and actions to take
- Watch rules allow ignoring specific files or entire directories within the watched tree.

It’s Demo Time

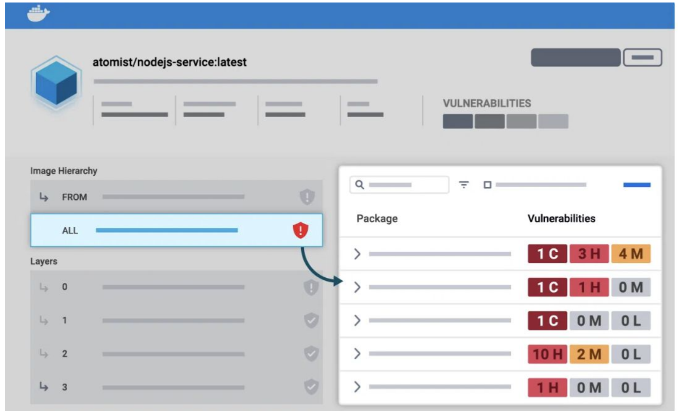
https://github.com/dockersamples/avatars

Container Security Monitoring for Developers

docker® scout

Secure Your Supply Chain at each Layer

- Unified view of securing your container development
- Includes a layer-by-layer view of dependencies, their known vulnerabilities, and recommended remediation paths.
- Designed with developers in mind
- 1st Class Citizen(integrated directly into Docker)
- Sits as a layer on top of the Docker ecosystem to help developers build and maintain a secure software supply chain



Package	Vulnerabilities
	1 C 3 H 4 M
	1 C 1 H 0 M
	1 C 0 M 0 L
	10 H 2 M 0 L
	1 H 0 M 0 L

"Spend less time search for and fixing vulnerabilities"

Docker Scout

Observability & Analysis

With one view, your application's direct and transitive dependencies from all layers are visible. This layer-based view not only makes remediation next steps clear, it also builds understanding of image composition.

Vulnerability Management

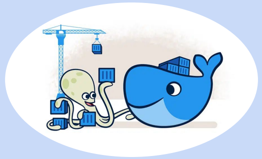
When a new CVE is released, Docker Scout uses your image's SBOM to check whether there's a positive correlation between your image and your CVE – so your recommendations are always up to date.

Integrated Remediation & Recommendation

Integrated recommendations are visible in Docker Desktop. Docker Scout recommends remediation options for base image updates, as well as dependency updates within your application code layers.

Docker Desktop

Full-Development Environments



Volume Management

Docker Compose V2

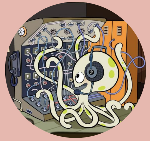
Support for VirtioFS

virtiofs

Docker Scout

docker scout

Docker Extensions



Support for Kubernetes

Dev Environments

Docker Desktop

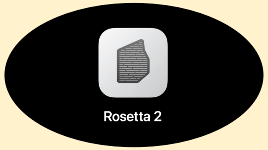
Support for Containerd

containerd

Support for Wasm

WA

x86/amd64 Binary Emulation



SBOM Indexing

Docker init

Compose Include

File Watch

Docker & Wasm – Better Together

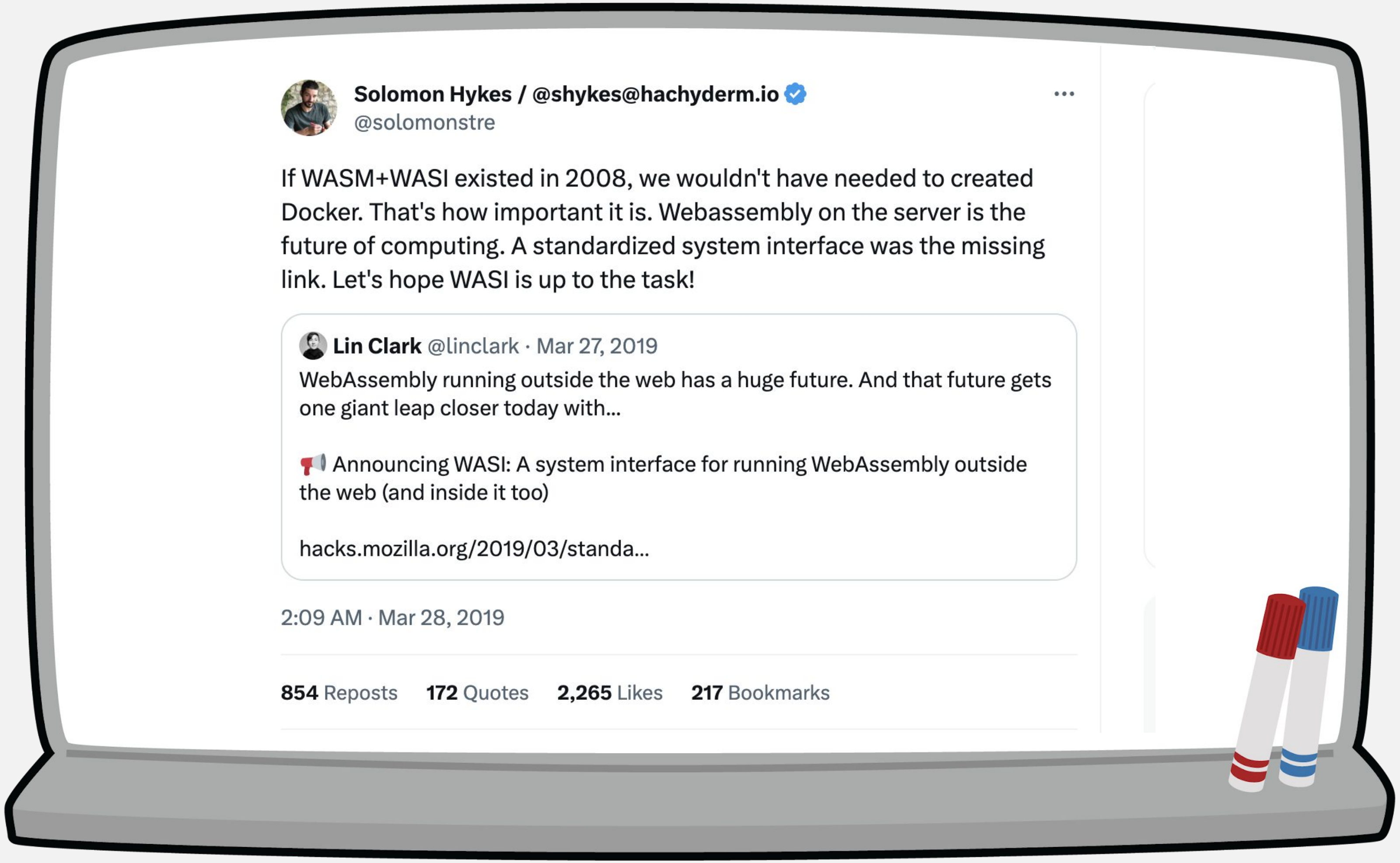
$$\begin{tikzcd}\boxed{\text{PHYSICAL HARDWARE}} \arrow[r] & \boxed{\text{VIRTUAL MACHINE}} \arrow[r] & \boxed{\text{CONTAINERS}} \arrow[r] & \boxed{\text{WEBASSEMBLY}}\end{tikzcd}$$

docker

WA

WEBASSEMBLY

docker



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!

Lin Clark @linclark · Mar 27, 2019
WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

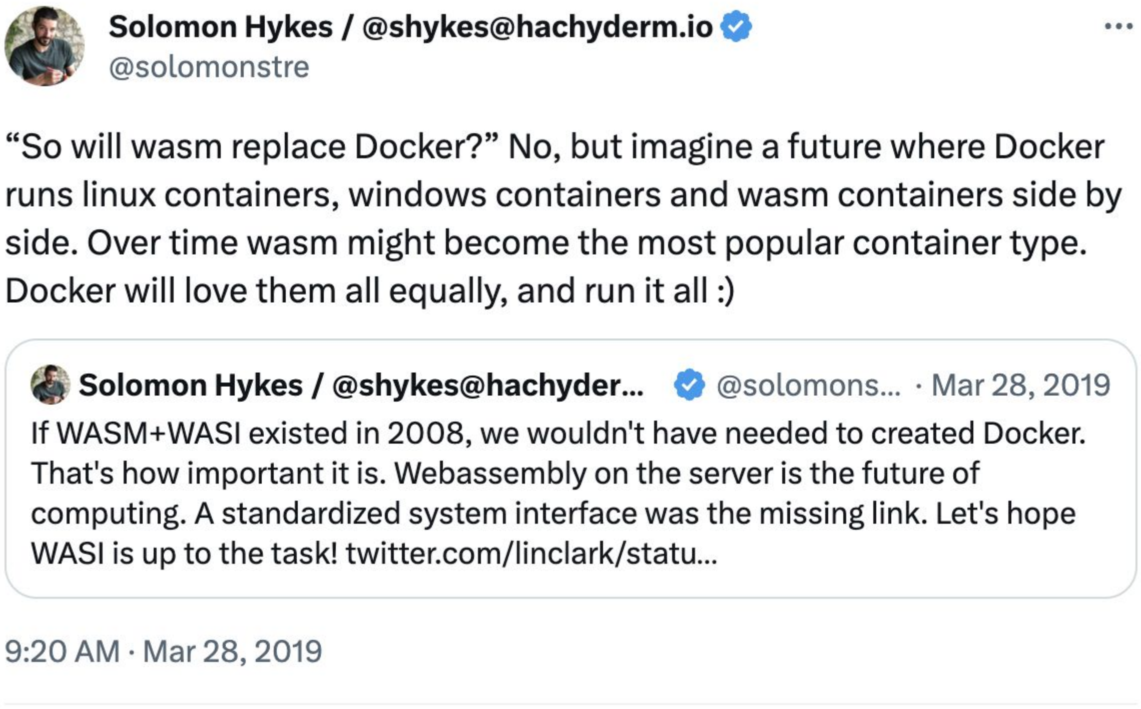
📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

hacks.mozilla.org/2019/03/standa...

2:09 AM · Mar 28, 2019

854 Reposts **172** Quotes **2,265** Likes **217** Bookmarks

docker



Solomon Hykes / @shykes@hachyderm.io
@solomonstre

"So will wasm replace Docker?" No, but imagine a future where Docker runs linux containers, windows containers and wasm containers side by side. Over time wasm might become the most popular container type. Docker will love them all equally, and run it all :)

> **Solomon Hykes / @shykes@hachyder...** @solomons... · Mar 28, 2019
> 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! twitter.com/linclark/statu...

9:20 AM · Mar 28, 2019

What is WebAssembly(Wasm)?

Web + Assembly $\longrightarrow$ Wasm

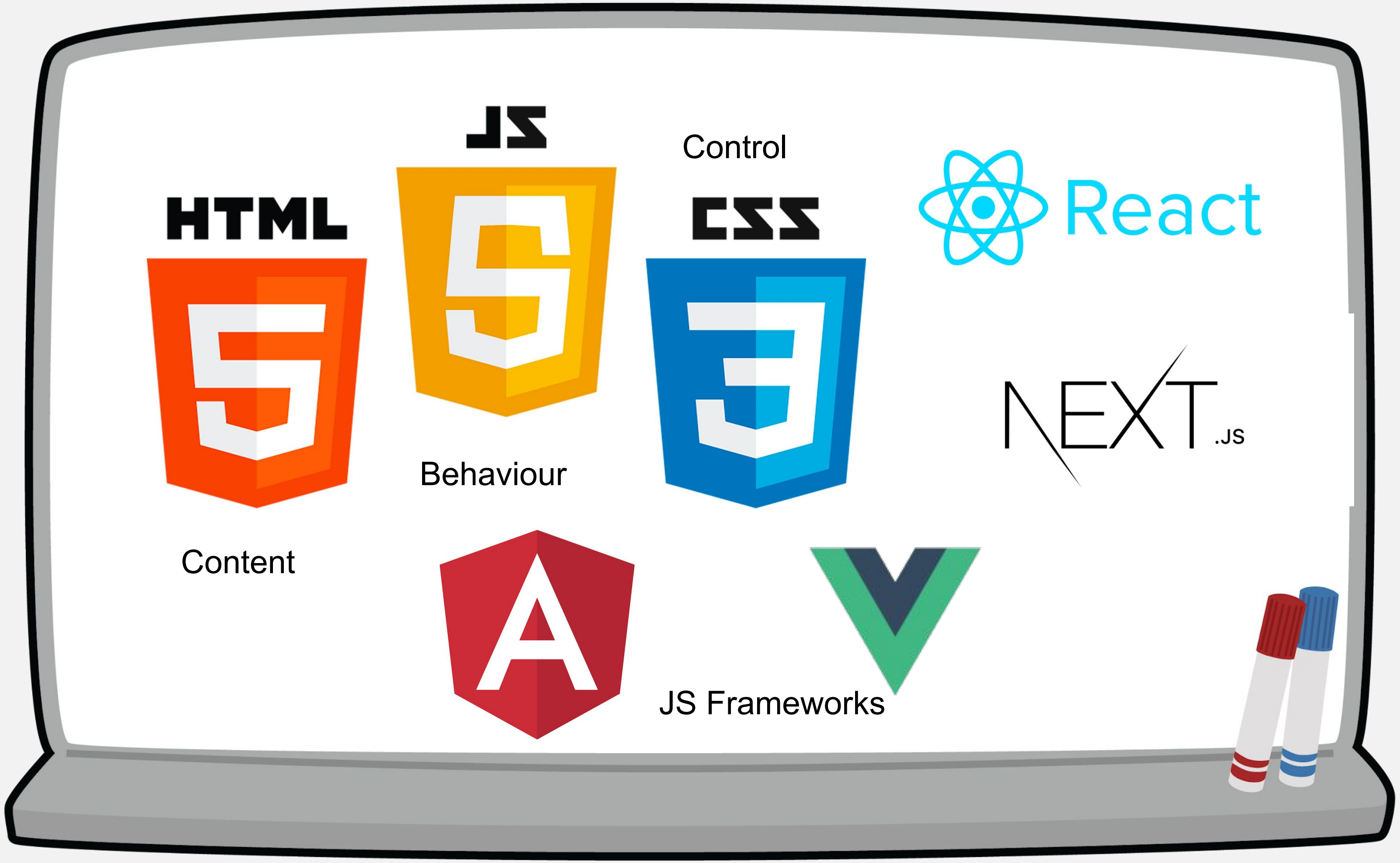
Web: interconnected system of public web pages

Assembly: Low-Level Programming Language

Wasm: Not an Assembly Language BUT a new binary-based Programming Language where code runs on a Web Browser

WA
WEBASSEMBLY

docker


JS
Control
HTML
CSS
React
Behaviour
NEXT.JS
Content
JS Frameworks

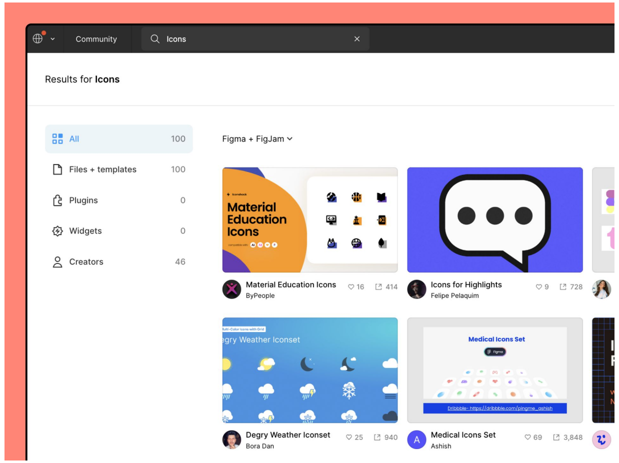
docker

Adobe used WASM to port Photoshop to the web

Uses the Emscripten compiler to convert Photoshop's C++ code to WASM. Emscripten is a compiler that takes code written in C/C++ and converts it to JavaScript, which can then be run in a web browser.

Uses Wasm in the browser and Docker to start an HTTP server

Figma



- Render the vector graphics that are used in Figma Designs
- Calculate complex algorithm that are used in Figma
- Allows users to create plugins that extends the app functionality

docker

Canva

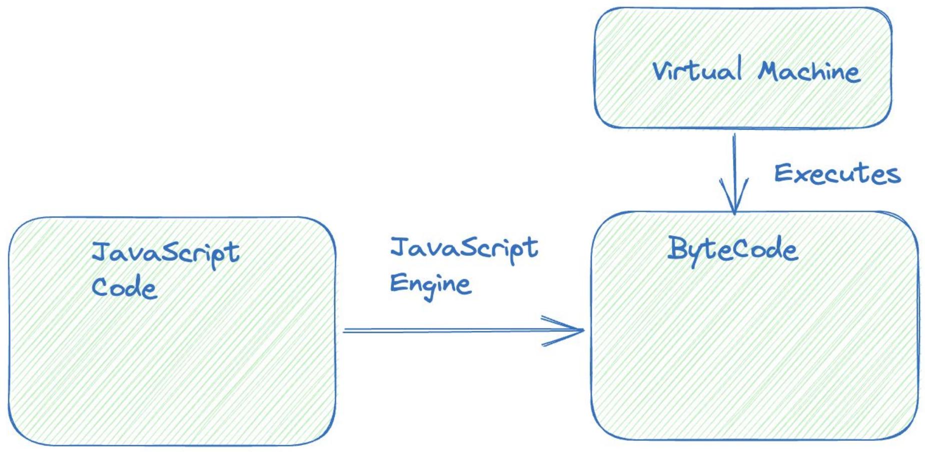
MINECRAFT
JAVA & BEDROCK

Instructions encoded in binary format

“Wasm is a **binary instruction format** for a **stack-based Virtual Machine**”

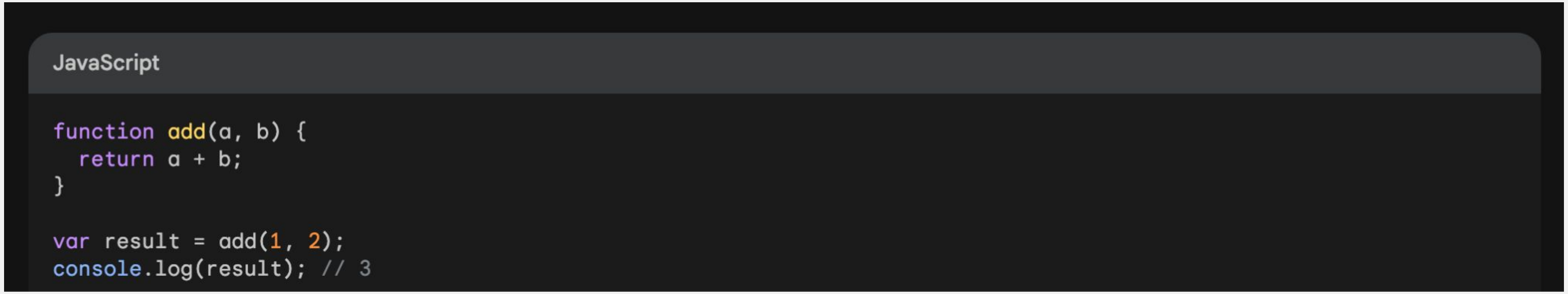
VM that uses a stack data structure to store data

Wasm	JavaScript
Wasm is binary-format	JavaScript is text-format
Wasm is compiled, wasm code is converted into machine code before it is executed	JavaScript is interpreted , code is interpreted line by line
Wasm is sandboxed(isolated from rest of the browser)	JavaScript is not sandboxed(has access to the entire browser)
If you're developing a high-performance, portable and secure application	If you're developing a simple application that doesn't require high performance



The JavaScript code is first converted into bytecode by the JavaScript engine.
The bytecode is then executed by the virtual machine.
The virtual machine is a program that understands the bytecode and can execute it.

When you write a program in JavaScript, the code is first converted into bytecode. Bytecode is a format that can be interpreted by the JavaScript engine in the browser. The JavaScript engine then executes the bytecode line by line.



JavaScript

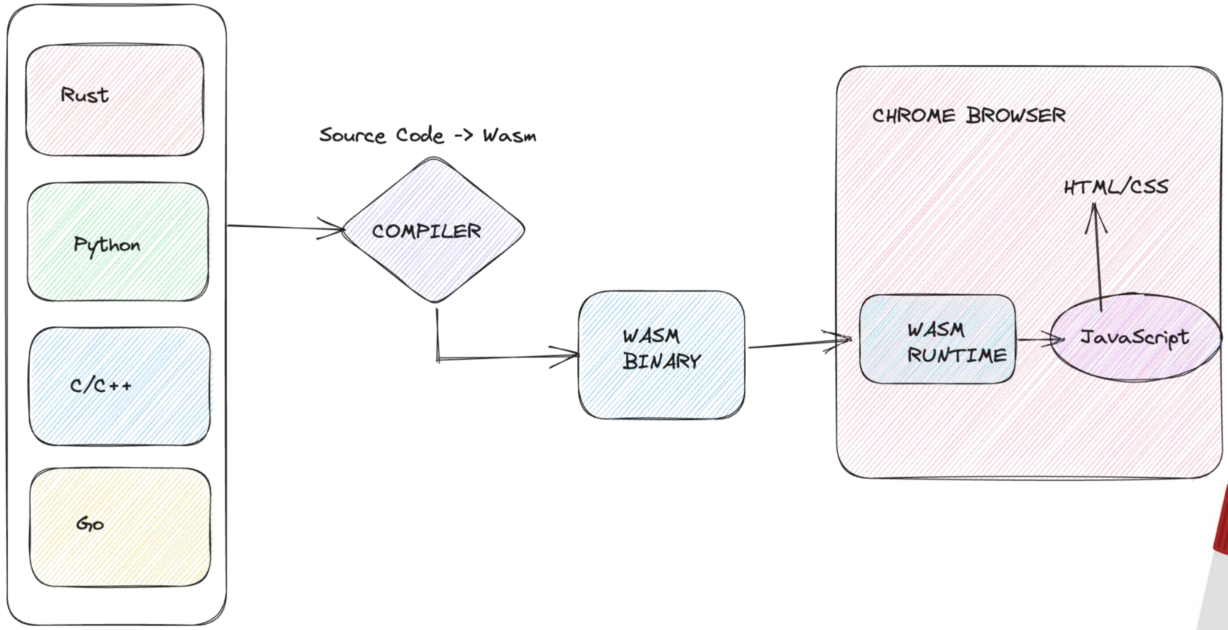
```javascript
function add(a, b) {
  return a + b;
}

var result = add(1, 2);
console.log(result); // 3
```

The bytecode for this program is as follows:

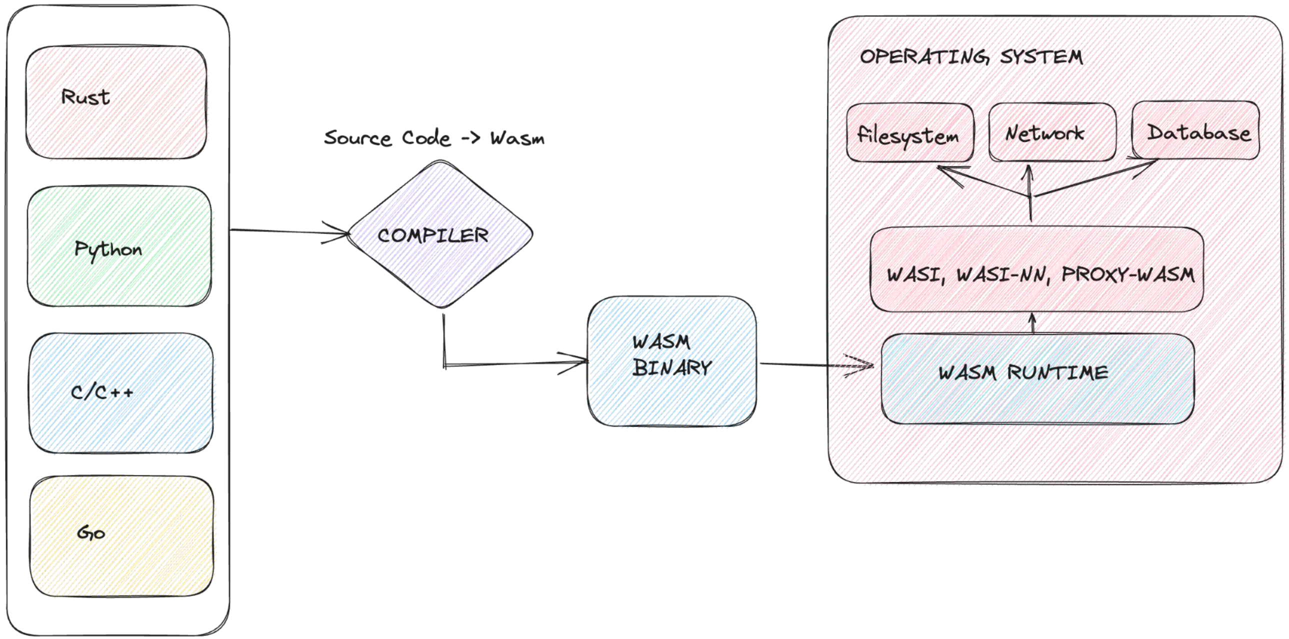
```
0001 load_arg0
0002 load_arg1
0003 add
0004 return
```

JavaScript is a programming language, and bytecode is a low-level intermediate language.

How Wasm works on Browser?



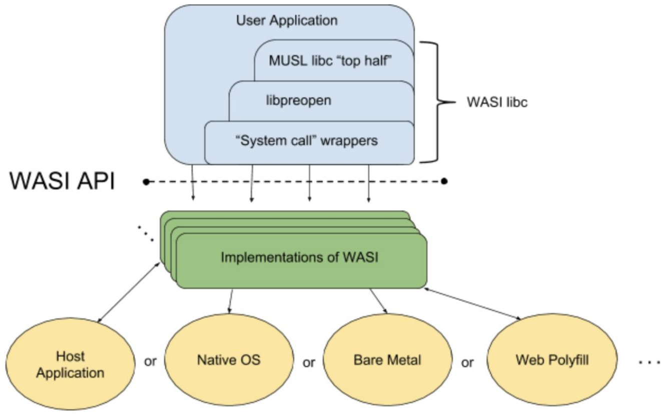
How Wasm works on Server?



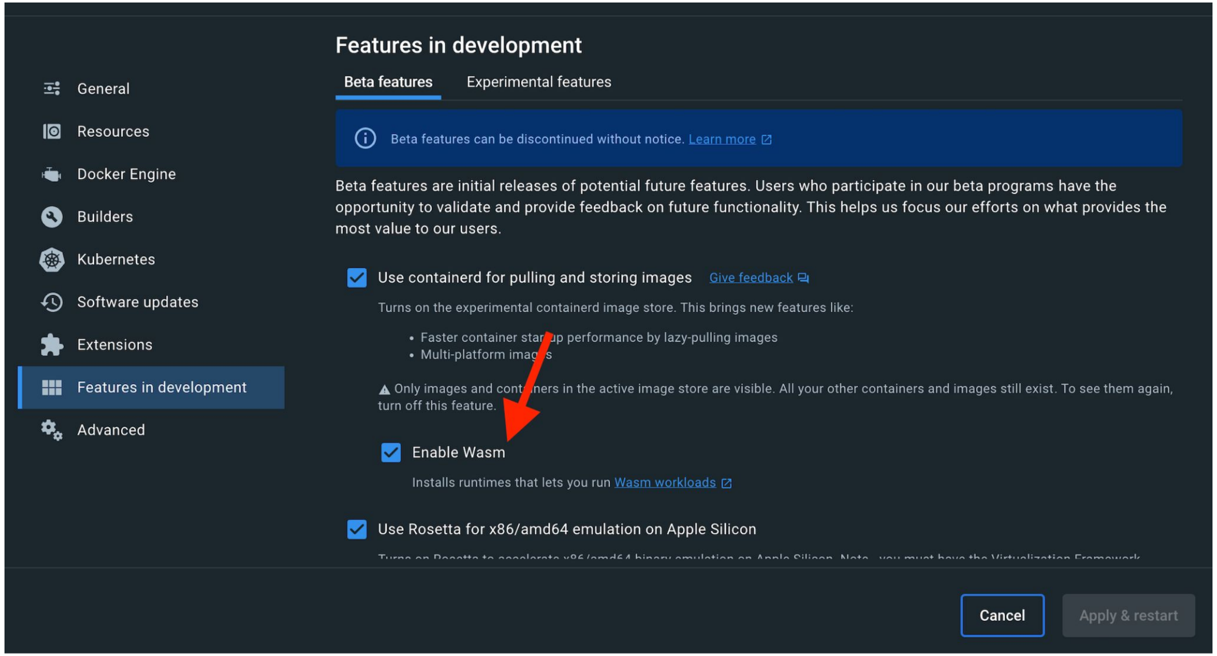
docker

What is WASI?

- WASI is a specification that defines how WASM code can interact with the host environment
- Provides a set of APIs that allow WASM code to access the browser's resources, such as files, network and timers



Wasm and Docker Desktop



Features in development

Beta features can be discontinued without notice. Learn more

Beta features are initial releases of potential future features. Users who participate in our beta programs have the opportunity to validate and provide feedback on future functionality. This helps us focus our efforts on what provides the most value to our users.

[✓] Use containerd for pulling and storing images Give feedback

Turns on the experimental containerd image store. This brings new features like:

- Faster container startup performance by lazy-pulling images
- Multi-platform images

⚠ Only images and containers in the active image store are visible. All your other containers and images still exist. To see them again, turn off this feature.

[✓] Enable Wasm

Installs runtimes that lets you run Wasm workloads

[✓] Use Rosetta for x86/amd64 emulation on Apple Silicon

Why Docker is supporting Wasm?

- Conquer the app complexity
- Learn and Develop Faster
- Collaborate and Innovate
-

HELM

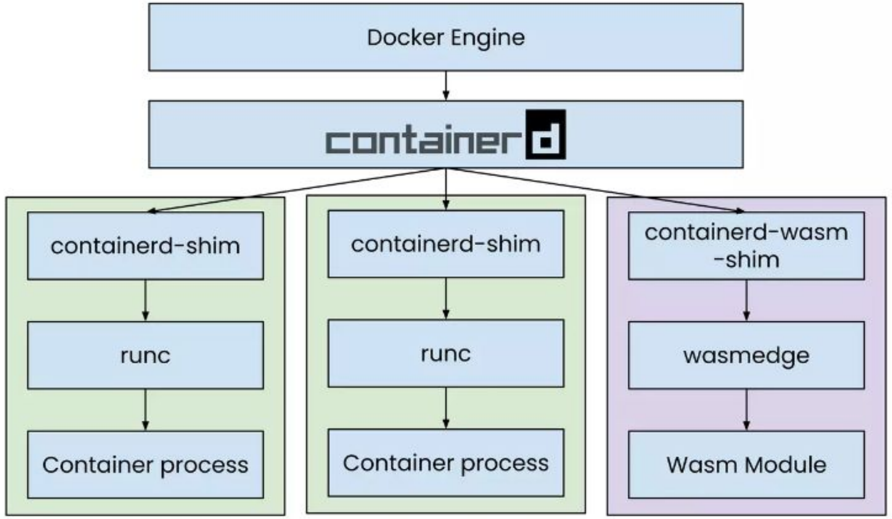
Generating SBOMs

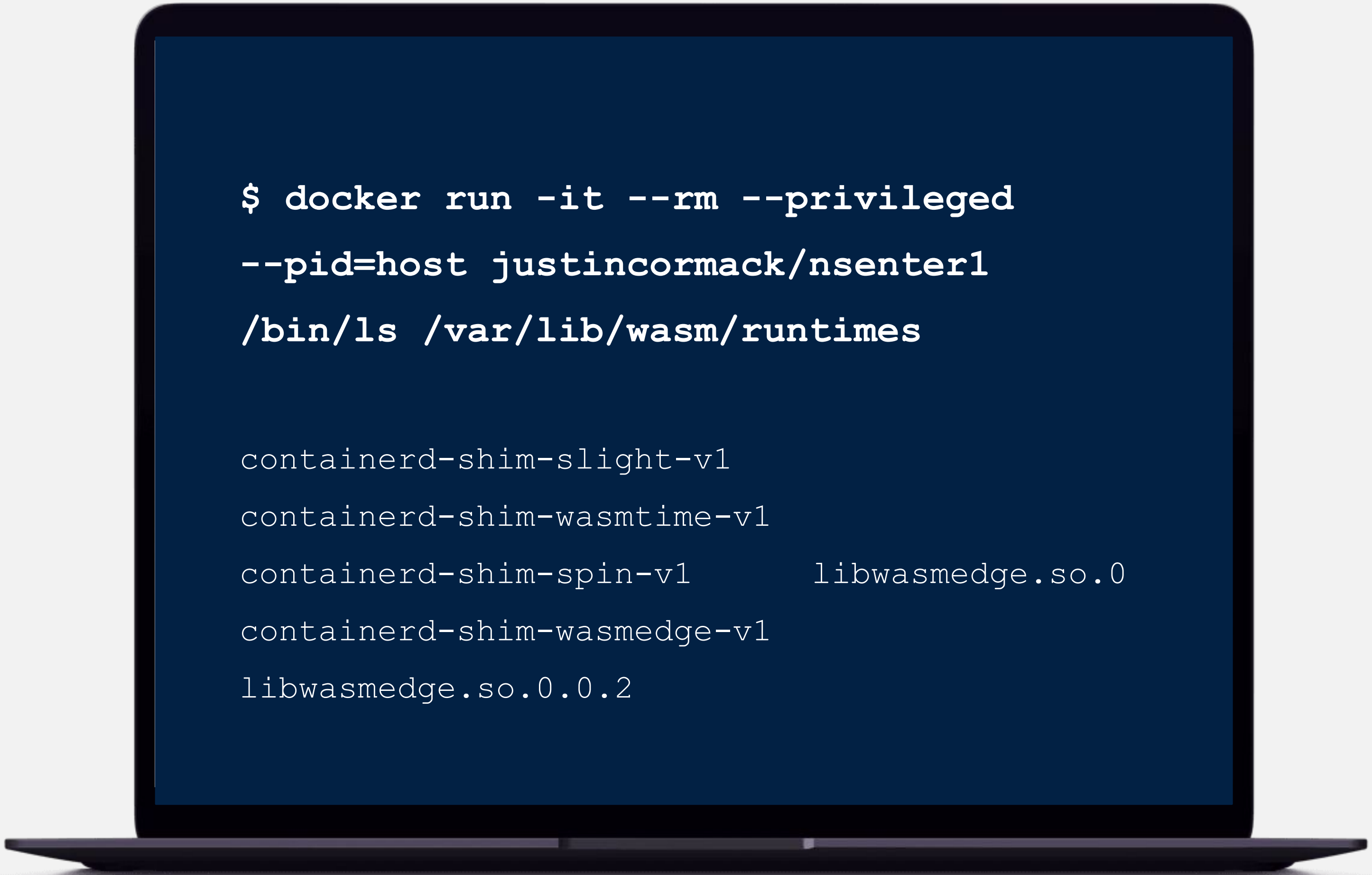
BuildKit

WA

WEBASSEMBLY

Wasm and Docker Desktop





```
$ docker run -it --rm --privileged
--pid=host justincormack/nsenter1
/bin/ls /var/lib/wasm/runtimes

containerd-shim-slight-v1
containerd-shim-wasmtime-v1
containerd-shim-spin-v1        libwasmedge.so.0
containerd-shim-wasmedge-v1
libwasmedge.so.0.0.2
```

Supported Wasm Runtimes on Docker Desktop

SPIN

WasmEdge

BYTECODE ALLIANCE

Wasmtime

Slight, DeisLab

Spin - A Serverless Wasm Runtime

- Spin is a serverless Wasm runtime that is designed to be easy to use and deploy.
- A good choice for running Wasm workloads that are event-driven or that need to be scaled horizontally.

WasmEdge - Flexibility and Control

- WasmEdge is a full-featured Wasm runtime that supports a wide range of features.

  A good choice for running Wasm workloads that require a high degree of flexibility and control.

Wasmtime

- Wasmtime is a Wasm runtime that is designed to be compatible with the WebAssembly

  Specification. A good choice for running Wasm workloads that need to be portable to different environments.

Slight

- A lightweight Wasm runtime that is designed to be fast and efficient.
- A good choice for running Wasm workloads that require high performance.

It’s Demo Time

https://github.com/collabnix/wasmlabs/blob/main/dockerdesktop/demo/README.md

Get Connected

https://launchpass.com/collabnix