

wasmedge in k8s

实现思路

- 安装wasmedge（要用脚本安装）
- 安装crun (OCI runtime)
 - The crun project has WasmEdge support baked in.
- 安装crio/containerd
- 配置crio/containerd使用crun作为runtime，并启动crio/containerd服务

- ```
[crio.runtime]
default_runtime = "crun"
```

```
[crio.runtime.runtimes.runc]
runtime_path = "/usr/lib/cri-o-runc/sbin/runc"
runtime_type = "oci"
runtime_root = "/run/runc"
The above is the original content

Add our crunw runtime here
[crio.runtime.runtimes.crun]
runtime_path = "/usr/bin/crun"
runtime_type = "oci"
runtime_root = "/run/crun"
```

- 下载k8s源码，编译并运行k8s（使用containerd）
  - `sudo CGROUP_DRIVER=systemd CONTAINER_RUNTIME=remote CONTAINER_RUNTIME_ENDPOINT='unix:///var/run/crio/crio.sock' ./hack/local-up-cluster.sh`

# wasmedge in k8s

## 实现思路

- 装载wasm程序的image
  - simply embed the WebAssembly bytecode file in a Linux container image
    - The memory footprint of the entire image of Linux OS and WasmEdge can be reduced to as low as **4MB**.
    - The **performance and security** of this approach **would not be** as great as running WebAssembly applications directly in crun or in a containerd shim.

```
FROM wasmedge/slim-runtime:0.10.1
ADD wasi_example_main.wasm /
CMD ["wasmedge", "--dir", " ./", "/wasi_example_main.wasm"]
```

```
FROM scratch
COPY hello.wasm /
CMD ["/hello.wasm"]
```

# wasmedge in k8s

## 实现思路

- 装载wasm程序的image
  - use the containerd shim

As we discussed, wrapping WebAssembly inside a Docker Linux container results in performance and security penalties. However, we cannot easily replace the OCI runtime ( `runc` ) in the Docker toolchain as well. In this chapter, we will discuss another approach to start and run WebAssembly bytecode applications directly from the Docker CLI.

- Coming soon

# wasmedge in k8s

## 验证

- crio没有验证成功
  - crio直接运行container没有成功，报错：image unknown
  - crio与k8s一起运行没有成功

```
~/Doc/P/f/kubernetes | #v1.25.3 *1 ?2 | sudo cluster/kubect1.sh get pod --all-namespaces
[sudo] password for leviyan:
NAMESPACE NAME READY STATUS RESTARTS AGE
default http-server 0/1 ContainerCreating 0 4h44m
kube-system coredns-567b6dd84-bs7bz 0/1 ContainerCreating 0 4h51m
```

```
Events:
 Type Reason Age From Message
 ---- -
 Warning FailedCreatePodSandBox 21m (x4 over 93m) kubelet Failed to create pod sandbox: rpc error: code = Unknown desc = error creating pod sandbox with name "k8s_coredns-567b6dd84-bs7bz_kube-system_f105a3ca-13ce-4abd-b1c7-4c83703cec0e_0": Error initializing source docker://k8s.gcr.io/pause:3.5: error pinging docker registry k8s.gcr.io: Get "https://k8s.gcr.io/v2/": dial tcp 108.177.125.82:443: i/o timeout
 Warning FailedCreatePodSandBox 16m (x39 over 4h30m) kubelet Failed to create pod sandbox: rpc error: code = Unknown desc = error creating pod sandbox with name "k8s_coredns-567b6dd84-bs7bz_kube-system_f105a3ca-13ce-4abd-b1c7-4c83703cec0e_0": Error initializing source docker://k8s.gcr.io/pause:3.5: error pinging docker registry k8s.gcr.io: Get "https://k8s.gcr.io/v2/": dial tcp 142.250.157.82:443: i/o timeout
 Warning FailedCreatePodSandBox 97s (x133 over 4h51m) kubelet Failed to create pod sandbox: rpc error: code = Unknown desc = error creating pod sandbox with name "k8s_coredns-567b6dd84-bs7bz_kube-system_f105a3ca-13ce-4abd-b1c7-4c83703cec0e_0": Error initializing source docker://k8s.gcr.io/pause:3.5: error pinging docker registry k8s.gcr.io: Get "https://k8s.gcr.io/v2/": dial tcp 142.251.8.82:443: i/o timeout
```

- 本地机的访问docker有点问题；云服务器没有虚拟化的能力，不能运行k8s



# wasmedge in k8s

## 验证

- containerd
  - 配置使用crun并运行wasm程序成功
  - [https://wasmedge.org/book/en/use\\_cases/kubernetes/cri/containerd.html](https://wasmedge.org/book/en/use_cases/kubernetes/cri/containerd.html)
  - 内容有错，编译k8s的命令参考：<https://raw.githubusercontent.com/second-state/wasmedge-containers-examples/main/kubernetes/containerd/install.sh>

```
~/Downloads: sudo ctr run --rm --runc-binary crun --runtime io.containerd.runc.v2 --label module=compat-smart docker.io/wasmedge/example-wasi:latest wasm-example /wasi_example_main.wasm 50000000

[sudo] password for leviyan:
Random number: -452635506
Random bytes: [90, 113, 55, 186, 166, 96, 87, 26, 74, 16, 217, 112, 254, 111, 101, 77, 225, 27, 5, 162, 198, 129, 212, 133, 149, 36, 190, 245, 204, 146, 36, 59, 233, 143, 113, 109, 118, 238, 128, 172, 101, 24, 145, 127, 142, 253, 68, 89, 126, 82, 70, 243, 67, 199, 24, 135, 52, 200, 255, 130, 118, 140, 105, 126, 149, 147, 254, 237, 64, 233, 160, 139, 89, 100, 103, 154, 29, 66, 219, 214, 187, 27, 79, 32, 60, 243, 181, 120, 110, 239, 8, 126, 199, 247, 24, 34, 96, 248, 72, 204, 46, 32, 214, 10, 201, 142, 110, 180]
Printed from wasi: This is from a main function
This is from a main function
The env vars are as follows.
PATH: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOME: /
The args are as follows.
/wasi_example_main.wasm
50000000
File content is This is in a file
```

# wasmedge in k8s

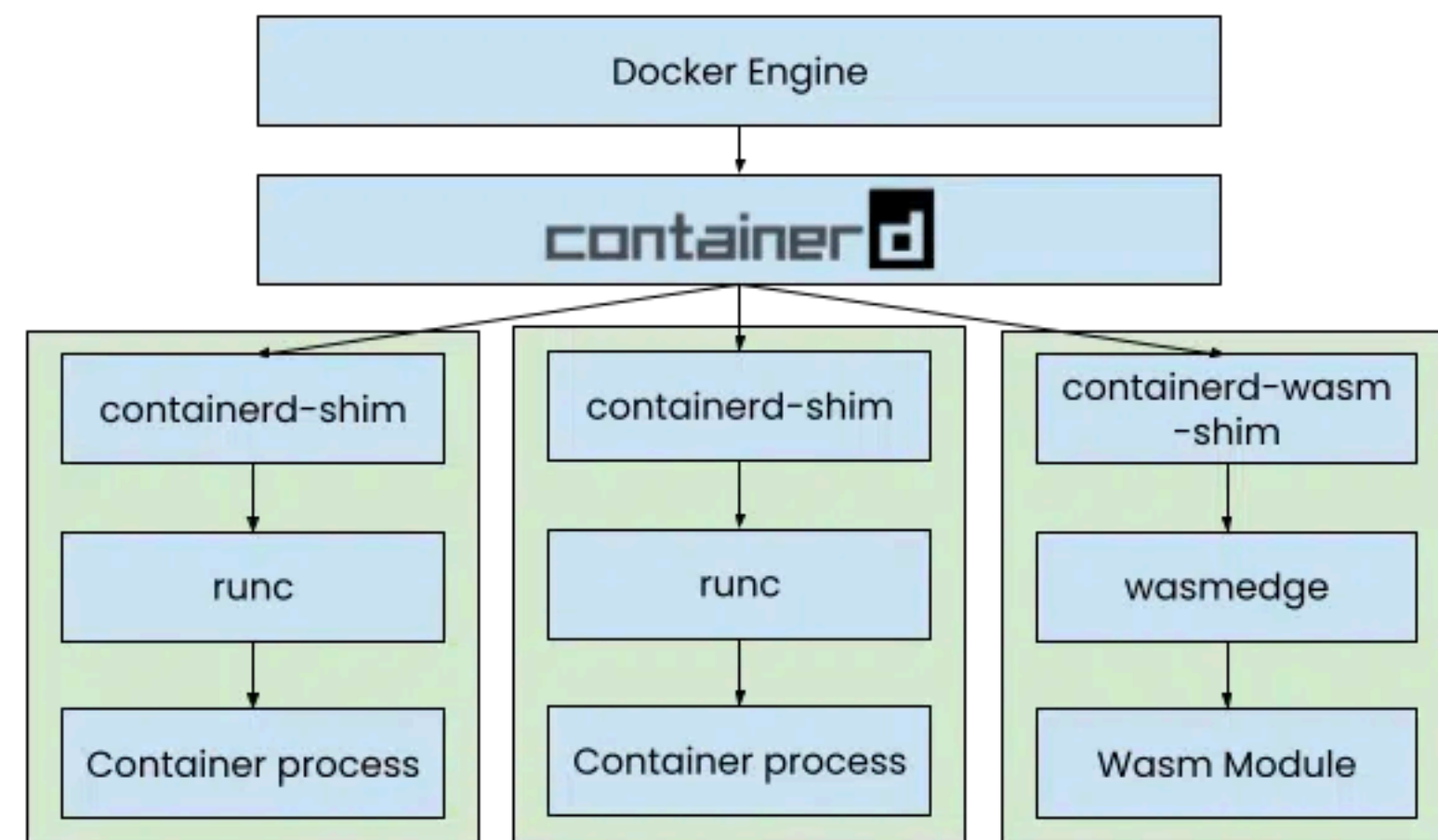
## 验证

- containerd
  - 使用k8s运行程序失败

```
Events:
 Type Reason Age From Message
 ---- -
 Warning FailedScheduling 5m32s default-scheduler no nodes available to schedule pods
 Warning FailedScheduling 5m30s default-scheduler 0/1 nodes are available: 1 node(s) had taint {node.kubernetes.io/not-ready:
}, that the pod didn't tolerate.
 Normal Scheduled 5m21s default-scheduler Successfully assigned kube-system/coredns-755cd654d4-gdnt6 to 127.0.0.1
 Warning FailedCreatePodSandBox 29s (x7 over 4m50s) kubelet Failed to create pod sandbox: rpc error: code = Unknown desc = failed to get
sandbox image "k8s.gcr.io/pause:3.5": failed to pull image "k8s.gcr.io/pause:3.5": failed to pull and unpack image "k8s.gcr.io/pause:3.5": failed to r
esolve reference "k8s.gcr.io/pause:3.5": failed to do request: Head "https://k8s.gcr.io/v2/pause/manifests/3.5": dial tcp 142.251.8.82:443: i/o timeout
```

# Docker+wasm

## 如何实现



- We collaborated with **WasmEdge** to create a **containerd shim**
- This shim extracts the **Wasm module** from the OCI artifact and runs it using the WasmEdge runtime.
- We added support to **declare the Wasm runtime**, which will enable the use of this **new shim**.
- Important note #2: This preview has the **containerd image store** enabled and cannot be disabled.

# 参考

- <https://www.docker.com/blog/docker-wasm-technical-preview/>
- [https://wasmedge.org/book/en/use\\_cases/kubernetes/container/crun.html](https://wasmedge.org/book/en/use_cases/kubernetes/container/crun.html)
- [https://wasmedge.org/book/en/use\\_cases/kubernetes/cri/containerd.html](https://wasmedge.org/book/en/use_cases/kubernetes/cri/containerd.html)
- [https://wasmedge.org/book/en/use\\_cases/kubernetes/kubernetes/kubernetes-containerd.html](https://wasmedge.org/book/en/use_cases/kubernetes/kubernetes/kubernetes-containerd.html)