

Benchmark

cri image

- benchmark测试代码
 - containerd/integration/benchmark/cri-image-test.go
 - 后续可以添加关于cri containerd test
 - 测试内容： cri image pull/status/remove (oci image, wasm module)
- benchmark测试脚本
 - containerd/benchmark-between-oci-and-wasm.sh
 - 后续添加关于cri containerd test
- cri image 测试结果

```
---- BENCH: BenchmarkWasmModuleInCri-2
cri_image_test.go:24: make sure the test wasm moduel doesn't exist in the cri plugin
cri_image_test.go:32: pull the wasm module into the cri plugin
cri_image_test.go:40: the wasm module should be seen in the cri plugin
cri_image_test.go:36: remove the wasm module from the cri plugin
PASS
ok      github.com/containerd/containerd/integration/benchmark 3.687s
```

```
BenchmarkImageInCri-2          1          1562564237 ns/op
---- BENCH: BenchmarkImageInCri-2
cri_image_test.go:83: make sure the test image doesn't exist in the cri plugin
cri_image_test.go:91: pull the image into the cri plugin
cri_image_test.go:102: the image should be seen in the cri plugin
PASS
ok      github.com/containerd/containerd/integration/benchmark 4.857s
```

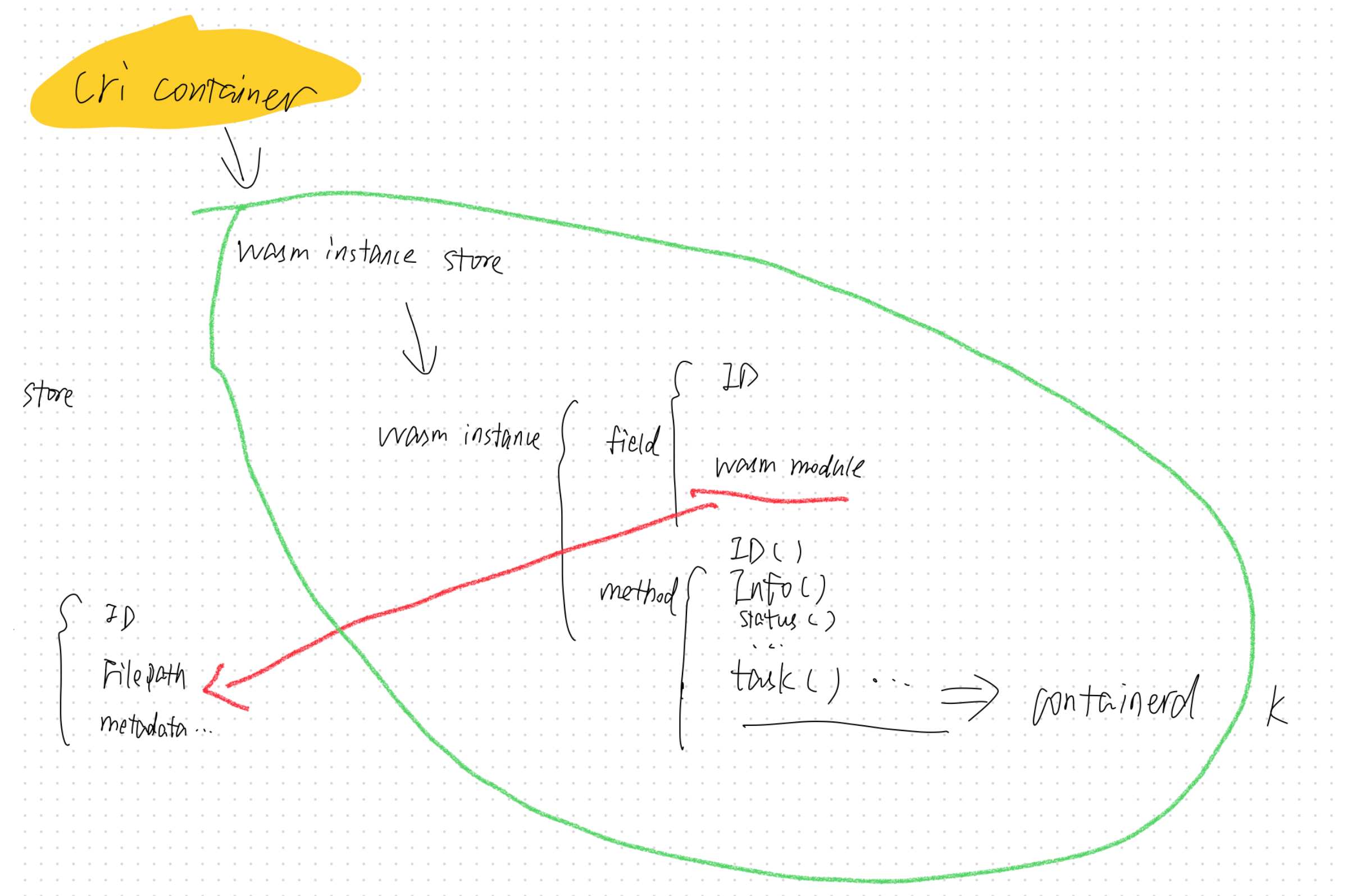
k8s调用的yaml文件格式

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  annotations:
    wasm-sandbox: 1      指明wasm sandbox
spec:
  runtimeHandler: runc
  containers:
    - name: wasm-name    wasm instance name
      image: wasm-module-name    wasm module name
      annotations:
        wasm.module.url: https://github.com/leviyanx/wasm-program-image/raw/main/wasi/wasi_example_main.wasm
      command: ["my-command"]
      args: ["args1", "args2", "args3...."]
```

●

Wasm instance的设计

- wasm instance代替container
- task/shim部分调用新实现的插件



问题

- pod与container使用的runtime handler是一样的（之前是runc）
 - pod使用的是pause容器
 - runwasi对于pause容器的处理就是做了一个没有任何操作的实例伪装处理

```
/// This is used for the "pause" container with cri and is a no-op instance implementation.  
pub struct Nop {  
    /// Since we are faking the container, we need to keep track of the "exit" code/time  
    /// We'll just mark it as exited when kill is called.  
    exit_code: Arc<ExitCode>,  
}
```

- exit_code 变量，表示容器的退出状态码和退出时间。
- start方法：启动一个容器实例，需要返回一个进程 ID，就返回当前进程的 ID
- kill方法：设置 exit_code 的值，表示容器已经退出

版本管理

- 从fork的origin/release1.6开始开发
- 分支
 - wasm_image: 开发wasm module相关功能
 - wasm_container: 开发wasm instance相关功能
 - wasm: 合并开发好的wasm功能

后续

- 集成测试代码 + 脚本
- 对接新service