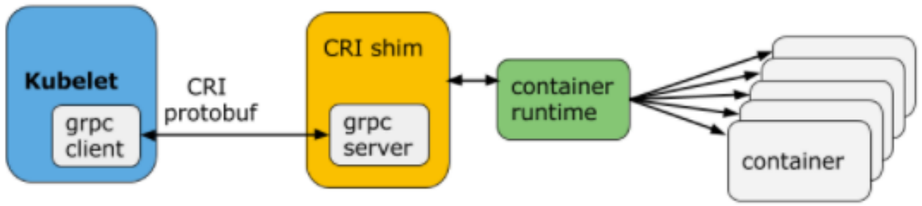
介绍CRI

概述

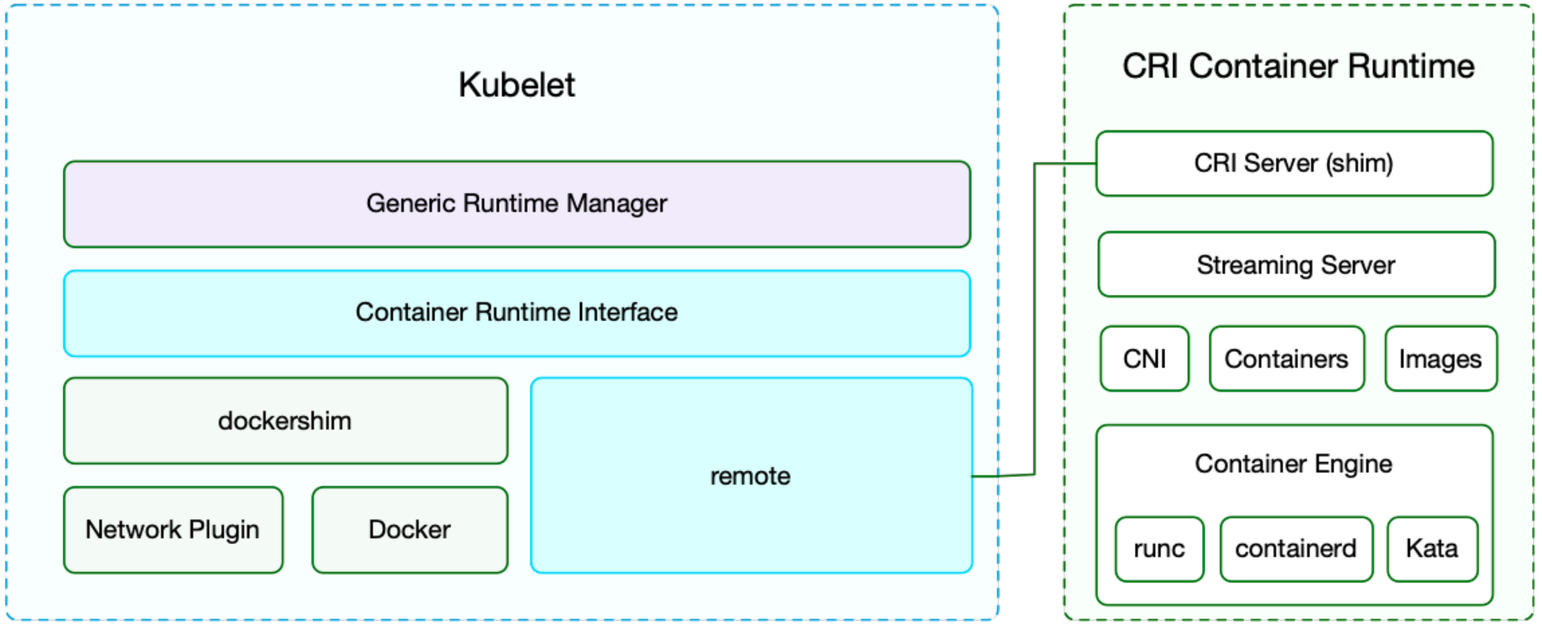
- 容器运行时插件（Container Runtime Interface，简称 CRI）
  - 是 Kubernetes v1.5 引入的容器运行时接口
  - 将 Kubelet 与容器运行时解耦
  - 将镜像管理和容器管理分离到不同的服务
  -



介绍CRI

概述

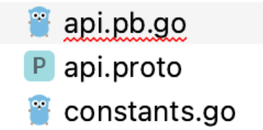
- 采用cri后，kubelet的结构
-



介绍CRI

具体

- CRI 基于 gRPC 定义了 **RuntimeService** 和 **ImageService** 等两个 gRPC 服务，分别用于**容器运行时**和**镜像**的管理。
  - 协议定义在**staging**/src/k8s.io/cri-api/pkg/apis/**runtime**/v1/**api.proto**中
  - 使用protocol buffers格式来定义




```
// Runtime service defines the public APIs for remote container runtimes
service RuntimeService {
    // Version returns the runtime name, runtime version, and runtime API version.
    rpc Version(VersionRequest) returns (VersionResponse) {}

    // RunPodSandbox creates and starts a pod-level sandbox. Runtimes must ensure
    // the sandbox is in the ready state on success.
    rpc RunPodSandbox(RunPodSandboxRequest) returns (RunPodSandboxResponse) {}
    // StopPodSandbox stops any running process that is part of the sandbox and
    // reclaims network resources (e.g., IP addresses) allocated to the sandbox.
    // If there are any running containers in the sandbox, they must be forcibly
    // terminated.
    // This call is idempotent, and must not return an error if all relevant
    // resources have already been reclaimed. kubelet will call StopPodSandbox
    // at least once before calling RemovePodSandbox. It will also attempt to
    // reclaim resources eagerly, as soon as a sandbox is not needed. Hence,
    // multiple StopPodSandbox calls are expected.
    rpc StopPodSandbox(StopPodSandboxRequest) returns (StopPodSandboxResponse) {}
```

```
// ImageService defines the public APIs for managing images.
service ImageService {
    // ListImages lists existing images.
    rpc ListImages(ListImagesRequest) returns (ListImagesResponse) {}
    // ImageStatus returns the status of the image. If the image is not
    // present, returns a response with ImageStatusResponse.Image set to
    // nil.
    rpc ImageStatus(ImageStatusRequest) returns (ImageStatusResponse) {}
    // PullImage pulls an image with authentication config.
    rpc PullImage(PullImageRequest) returns (PullImageResponse) {}
    // RemoveImage removes the image.
    // This call is idempotent, and must not return an error if the image has
    // already been removed.
    rpc RemoveImage(RemoveImageRequest) returns (RemoveImageResponse) {}
    // ImageFSInfo returns information of the filesystem that is used to store images.
    rpc ImageFsInfo(ImageFsInfoRequest) returns (ImageFsInfoResponse) {}
}
```

介绍CRI

具体

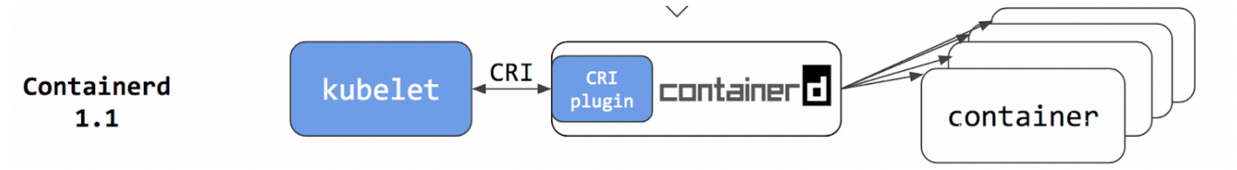
- Kubelet 作为 CRI 的客户端 (gRPC client)
  - pkg/kubelet/cri 实现了cri client
    - remote - image, runtime
    - stream - Exec、PortForward 和 Attach
- 容器运行时则需要实现 CRI 的服务端（即 gRPC server，通常称为 CRI shim）。

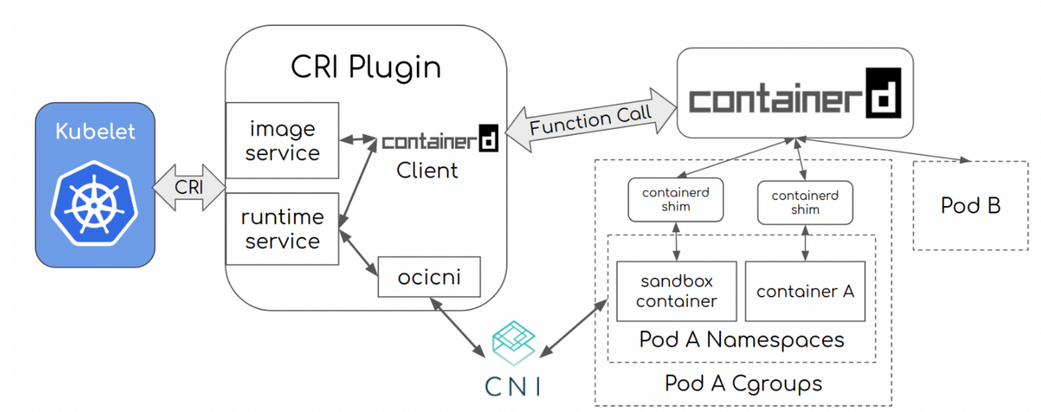
```
    internalapi "k8s.io/cri-api/pkg/apis"
    runtimeapi "k8s.io/cri-api/pkg/apis/runtime/v1"
    "k8s.io/klog/v2"
    "k8s.io/kubernetes/pkg/features"
    "k8s.io/kubernetes/pkg/kubelet/util"
    "k8s.io/kubernetes/pkg/probe/exec"
    utilexec "k8s.io/utils/exec"
)

// remoteRuntimeService is a gRPC implementation of internalapi.RuntimeService.
type remoteRuntimeService struct {
    timeout       time.Duration
    runtimeClient runtimeapi.RuntimeServiceClient
    // Cache last per-container error message to reduce log spam
    logReduction *logreduction.LogReduction
}
```

介绍CRI

具体

- 以containerd为例
  - 

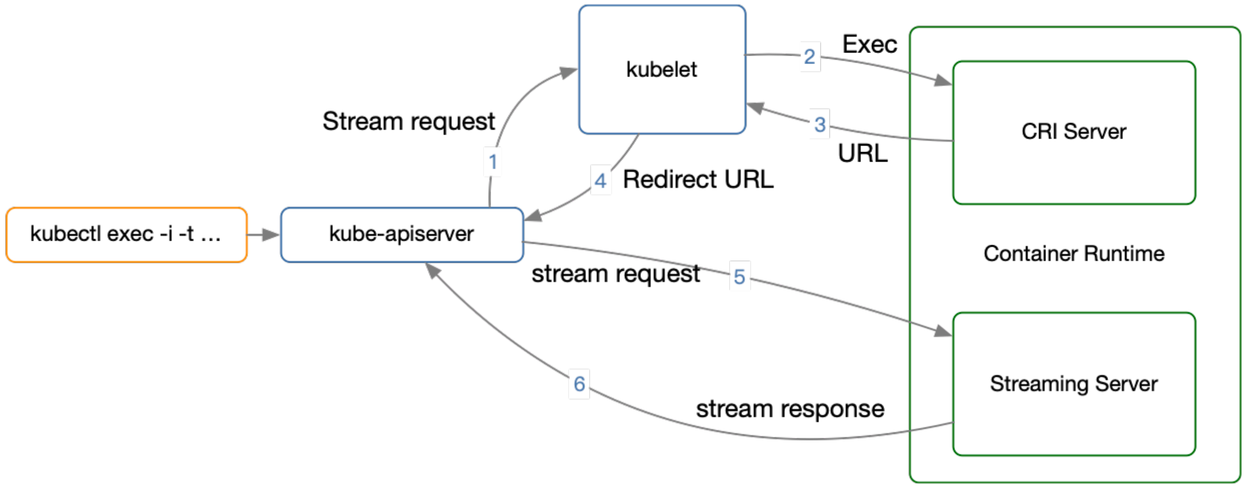
  - 

  - Containerd 内置的 CRI 插件实现了 Kubelet CRI 接口中的 Image Service 和 Runtime Service，通过内部接口管理容器和镜像，并通过 CNI 插件给 Pod 配置网络。

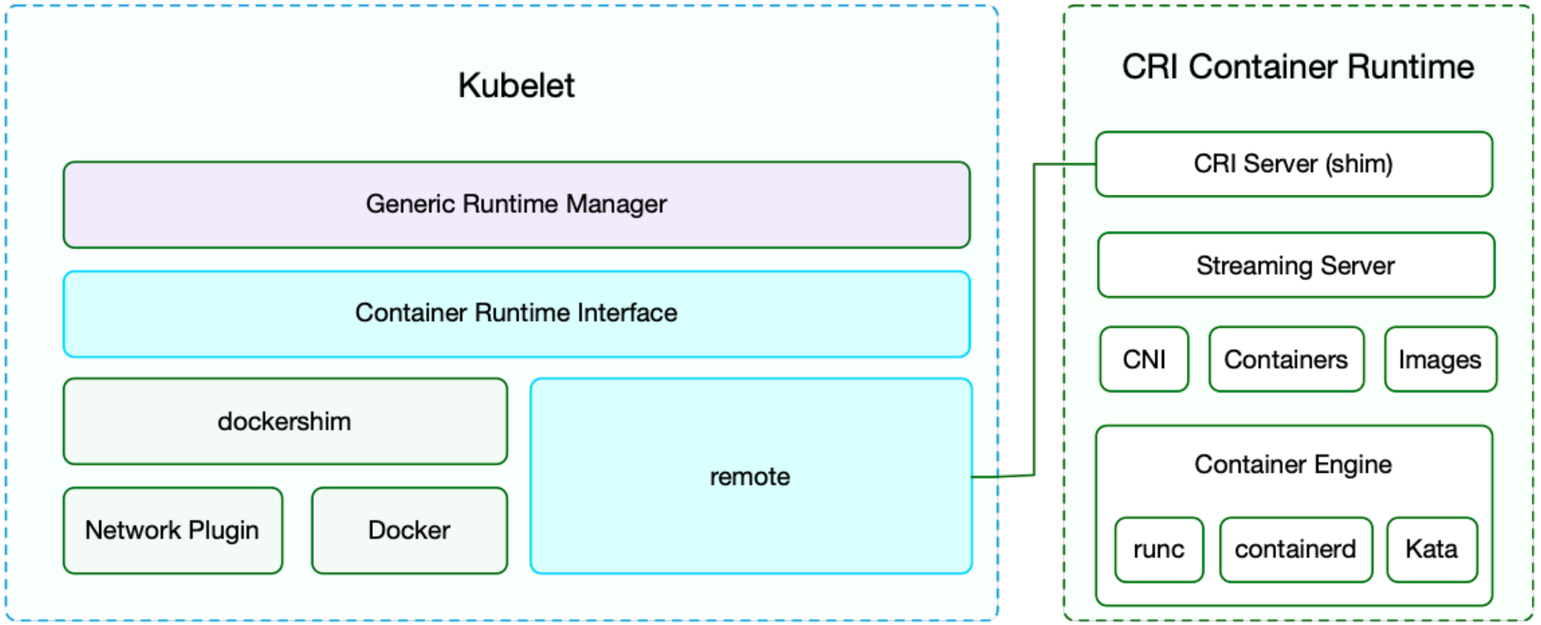
CRI

开发新的容器运行时

- 开发新的容器运行时只需要实现 CRI 的 gRPC Server，包括 RuntimeService 和 ImageService。
- 对于 Streaming API（Exec、PortForward 和 Attach），CRI 要求容器运行时返回一个 streaming server 的 URL 以便 Kubelet 重定向 API Server 发送过来的请求。



制作新的container runtime



参考

- **https://kubernetes.feisky.xyz/extension/cri**
- https://kubernetes.io/zh-cn/docs/concepts/architecture/cri/