

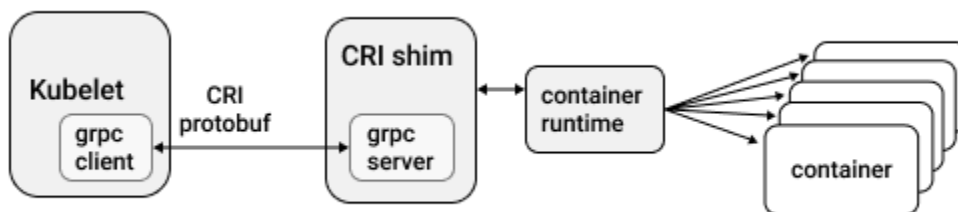
K8S 31: Kubernetes. Advanced. Runtime Environment

Описание:

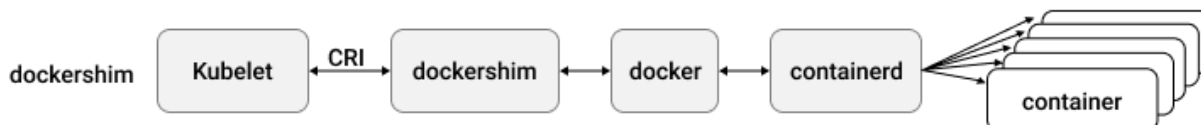
Как правило, для запуска контейнеров в Kubernetes используется Docker, как наиболее распространенное решение для запуска контейнеров. Но, на самом деле, Kubernetes поддерживает и другие системы для запуска контейнеров (container runtime) — реализовано это благодаря container-shim плагинам, которые реализуют интерфейс container-runtime. Некоторые из runtime:

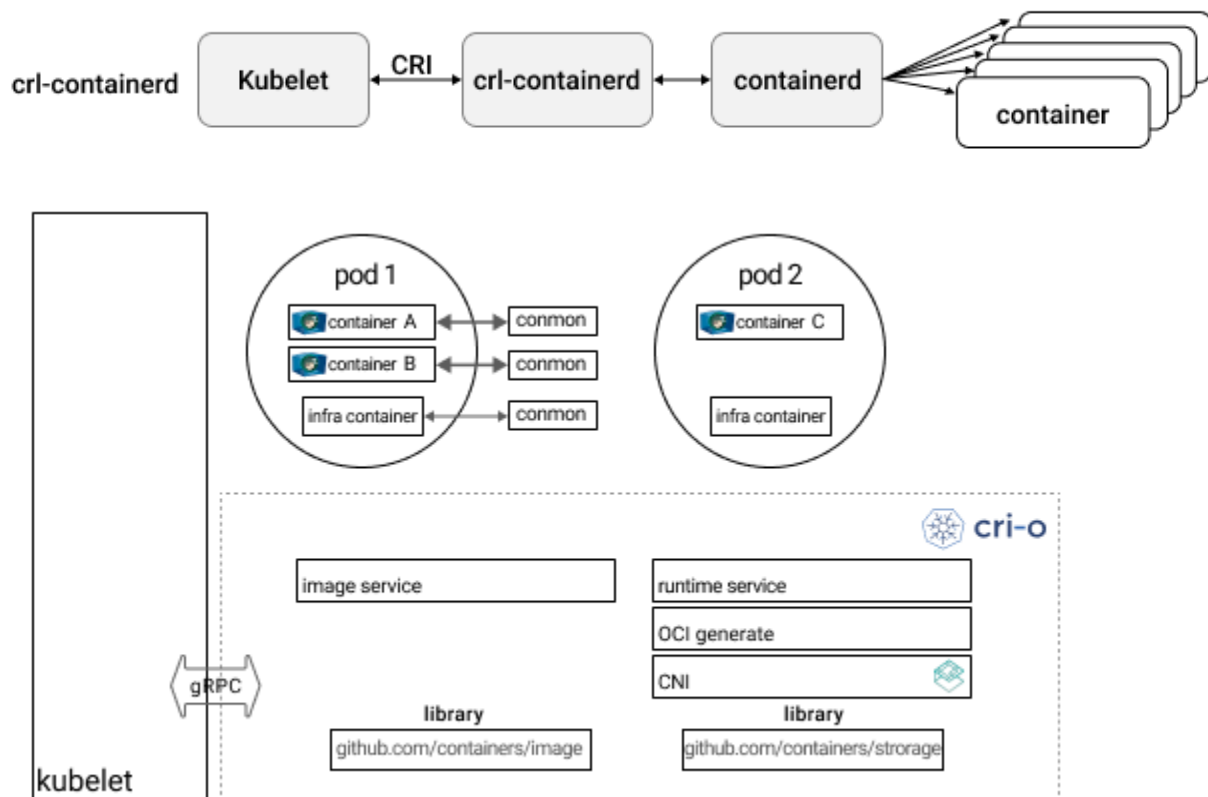
- Docker — думаю, не требуется рассказывать про этот индустриальный стандарт контейнеризации?
- rkt — создана компанией CoreOS Container Runtime, с идеей Pod в уме, что делает систему очень близкой по идеологии к K8s.
- cri-o — легковесная Container Runtime, созданная специально под Kubernetes.
- lxc — контейнеризация на Linux, созданная еще до того, как контейнеризация стала крутой (хотя как раз Docker в первых версиях использовал LXC для запуска).
- И многие другие (даже kvm!).

Схема взаимодействия выглядит следующим образом:



По аналогии с уже обсужденным CSI (Container Storage Interface), в Kubernetes используется CRI - Container Runtime Interface, который реализует универсальный (с учетом наличия плагина) shim, позволяющий Kubernetes общаться с Container Runtime. CRI реализует два сервиса — ImageService и RuntimeService. Первый отвечает за работу с образами, второй — за жизненный цикл контейнеров.





В большинстве случаев Docker в роли Container Runtime будет достаточен вам для любых нужд, но для знакомства мы попробуем с вами поработать с другим решением. А уж со стороны Kubernetes разницы в Runtime вы вообще чувствовать не должны — абстракции возьмут все на себя.

Полезные ссылки:

- [Container runtimes \(official docs\)](#)
- [LXE \(github\)](#)
- [Kubevirt \(github\)](#)
- [Introducing Container Runtime Interface \(CRI\) in Kubernetes \(official blog\)](#)
- [Контейнер – на конвейер: CRI-O теперь по умолчанию в OpenShift Container Platform 4 \(habr\)](#)
- [Kubernetes Container Runtimes](#)
- [CRI-O — альтернатива Docker для запуска контейнеров в Kubernetes \(habr\)](#)

Задание:

1. Запустите манифесты из задания про StatefulSet в вашем minikube кластере, если он еще не запущен.

2. Выведите список подов, созданных StatefulSet (команды и вывод сохранить).
3. Перезапустите minikube кластер с использованием cri-o в роли Container Runtime (команды и вывод сохранить).
4. Выведите список подов, созданных StatefulSet (команды и вывод сохранить).
5. Зайдите в виртуальную машину minikube и выведите список запущенных контейнеров (команды и вывод сохранить).
6. На проверку отправьте сохраненные команды и их выводы.