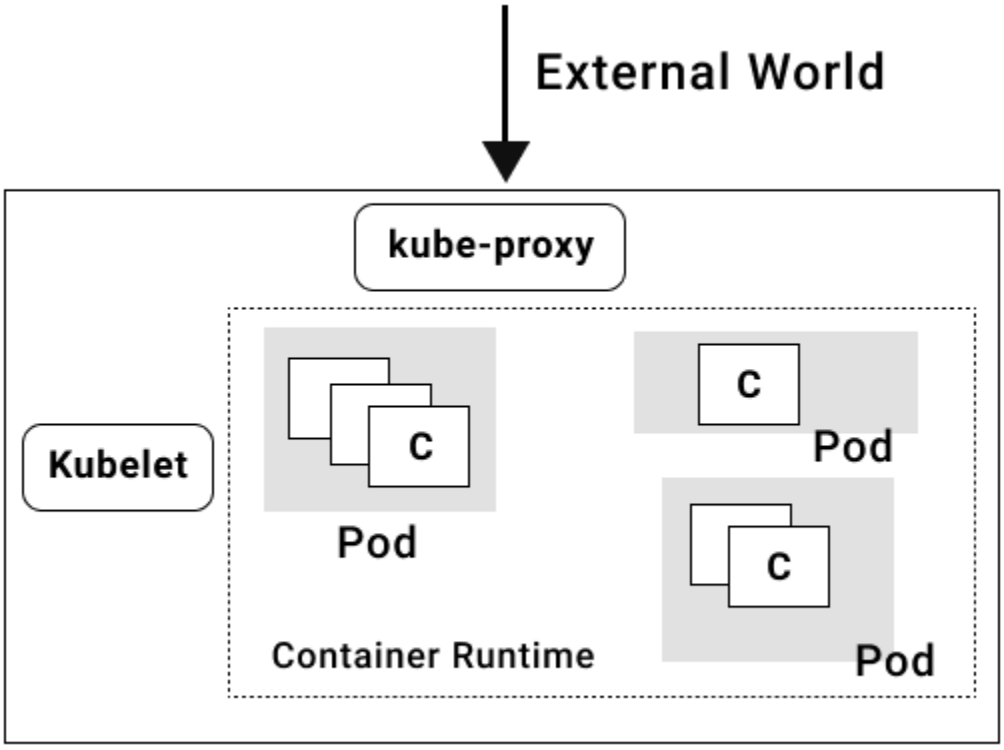
KUB 08: Kubelet & Kube-proxy

Описание:

В в седьмом задании мы с вами изучили Control Plane в Kubernetes и изучили основные компоненты, которые запускаются на мастер узлах. В этом задании же мы более детально обсудим работу worker nodes, на которых запускается полезная нагрузка.



На воркерах запускается три основных компонента:

- container runtime,
- kubelet,
- kube-proxy.

Container runtime

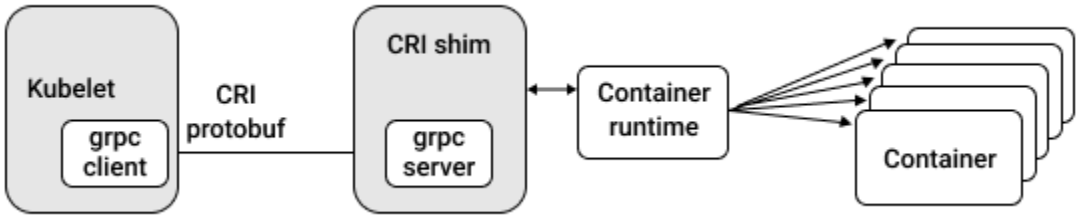
Container runtime — это совокупность инструментов для запуска контейнеров или среда исполнения контейнеров. В качестве таких инструментов может быть применен docker, cri-o, containerd и т.д.

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

- Docker — думаю, не имеет смысла расписывать про этот индустриальный стандарт контейнеризации? (В настоящее время объявлен как deprecated и к концу 2021 года перестанет устанавливаться по-умолчанию)
- rkt — создана компанией CoreOS Container Runtime, с идеей Pod в уме, что делает систему очень близкой по идеологии к K8s (На данный момент является deprecated)
- 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).
- containerd - проект по отделению среды исполнения от прочих компонентов docker. Является проектом CNCF и, на сегодняшний день, является самое перспективным с точки зрения поддержки и развития (продвигается CNCF как среда запуска по-умолчанию)
- gVisor - runtime от компании Google. Согласно документации "gVisor - это ядро приложения для контейнеров... и реализует Linux через Linux". Среда позволяет эмулировать вызовы ядра Linux и среда контейнеризации общается не с реальным ядром ОС, а неким программным эмулятором (что является более безопасным)
- И многие другие (даже kvm!).

Kubelet

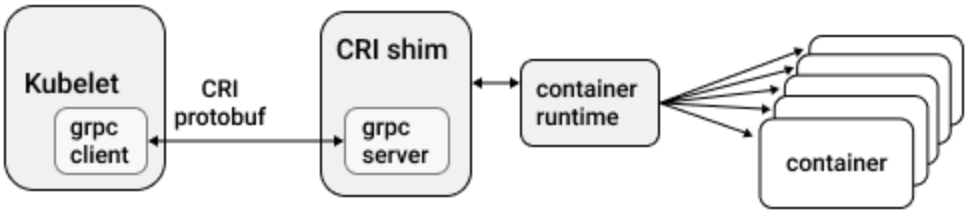
Агент кластера запускается на всяком рабочем узле, соединяется с мастером и работает с ним по API. Агент кластера получает задачи на выполнение от мастера. Кроме того, он описывает мастеру статус узла и контейнеров. Помимо мастера агент кластера работает с container runtime для запуска контейнеров, используя CRI (container runtime interface).



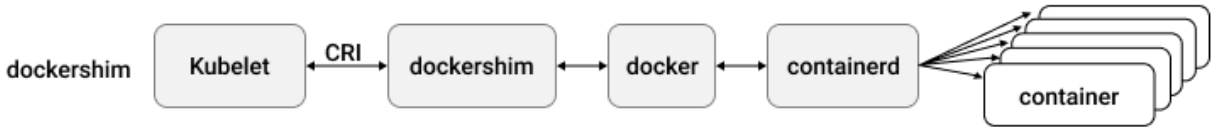
Через специальные плагины могут быть использованы различные реализации container runtime для запуска полезной нагрузки.

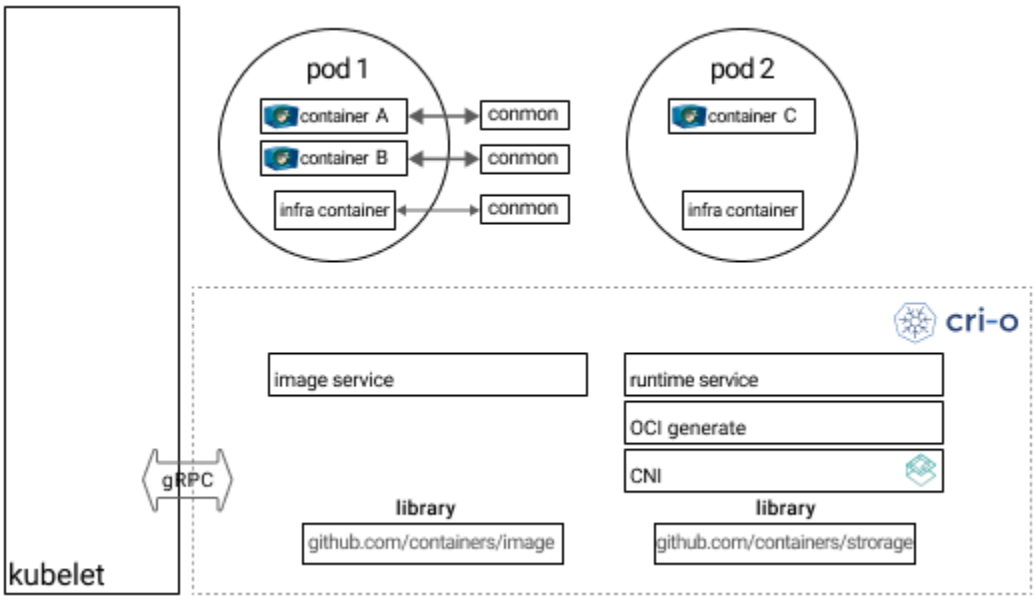
kube-proxy

Это сетевой прокси, запускаемый на всяком рабочем узле. Связывается с API-сервером для получения событий на создание/удаление Service, добавляет маршруты в таблицу маршрутизации и делает многое другое, разрешая контейнерам на разных узлах общаться друг с другом. Ещё этот элемент организовывает доступ к приложениям со стороны внешних пользователей.
Схема взаимодействия выглядит таким образом:



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 поработать с отличающимся решением. А уж со стороны Kubernetes разницы в Runtime вы совсем чувствовать не должны — абстракции возьмут все на себя.

Итак, давайте подведем небольшой итог и посмотрим еще раз как работают worker nodes:

1. На ноде запускается kubelet, который подключается к API Kubernetes и ждет заданий на запуск.
2. После получения задания на запуск пода, kubelet отправляет запрос локальному runtime - например, docker, а тот в свою очередь запускает указанный контейнер.
3. kube-proxy так же подключается к API Kubernetes и ждет события на создание или обновление ресурсов service, которые заставляют его настраивать iptables или ipvs для проброса портов на требуемые поды.

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)
- Nodes (official docs)
- Introduction to Kubernetes Architecture

Задание:

1. Разверните кластер kubernetes на трех нодах, используя конфиг kubespray, подготовленный в 3ем задании:
- В control plane должны находиться ноды node1 и node2
- Etcd должен быть запущен на нодах node1, node2, node3
- В качестве container runtime выберите containerd
2. Запустите под alpine с командой ping 8.8.8.8 в namespace default
3. Отправьте задание на проверку.