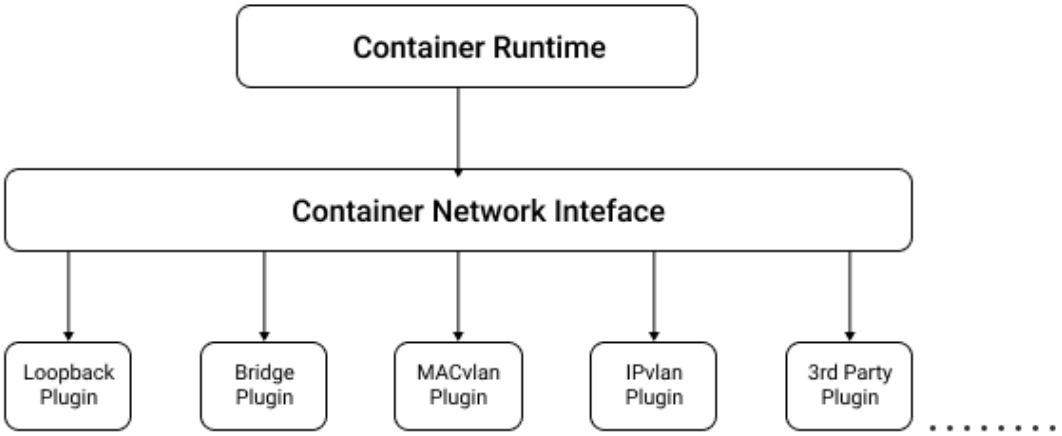
K8S 27: Kubernetes. Advanced. Networking

Описание:

Чтобы контейнеры могли общаться друг с другом нужен IP адрес для каждого контейнера. Нужна возможность общаться с внешним миром и доступность для внешних пользователей.
Есть два типа спеков на сети для контейнеров:

- CNI - Container Network Interface (CoreOS)
- CNM - Container Network Model (Docker)

Kubernetes использует CNI для назначения IP адресов подам.



При помощи ОС Container Runtime создает изолированную сеть для каждого контейнера. В Linux это обычно Network Namespace. Это пространство можно использовать с другими контейнерами или ОС.
Важно: Внутри POD контейнеры разделяют сетевое пространство и могут общаться друг с другом через localhost.
В кластере поды могут быть распределены на разные узлы (а вот контейнеры внутри пода - неделимы). Как обеспечить сетевую связность? Kubernetes не разрешает использование NAT при взаимодействии контейнеров. Получить это можно двумя способами:

- маршрутизируемая сеть между контейнерами, узлами настраиваемая заранее вручную или автоматически (например как в GCE)
- SDN (Software-defined networks), такие как Flannel, Weave, Calico и другие

Ручная настройка сети между всеми узлами кластера и распределение IP адресов для всех контейнеров очень сложная задача. Постоянно вести учет всех занятых и свободных адресов для сотен контейнеров просто невозможно. Многие облачные провайдеры предоставляют автоматически настраиваемую сеть. В таком случае остается только следить за количеством свободных IP адресов и не допускать коллизий. Однако это все еще оставляет много ручной работы. SDN призван решить данную проблему. В этом случае сеть работает в автоматическом режиме, IP адреса и маршруты настраиваются самостоятельно.
Роль CNI во всей этой схеме - это предоставление интерфейса общения между Kubernetes и сетевым решением, при котором Kubernetes не нужно подстраиваться под какое-то решение, наоборот используя стандартизацию мы имеем единый интерфейс, который может быть использован для связи kubernetes и различных реализаций CRI.
Сетевых решений для Kubernetes, на самом деле, довольно много - начиная от Flannel, который считается самым простым в плане реализации, заканчивая интеграцией с OpenVSwitch, что предоставляет огромные возможности организации связанности с уже существующей инфраструктурой
Различаются они как подходом к реализации сети (через bridge, IPVS, overlay), возможностями (сетевые политики, алгоритмы балансировки трафика на сервисах) и удобством установки (Calico vs Kube-router)
Как правило, адреса в кластере делятся на 2 сектора - адреса для Service и для Pods. Сделано это для того, чтобы визуально мы могли их различать

Полезные ссылки:

- Cluster Networking (official docs)
- Container Network Interface - networking for Linux containers (github)
- Introducing Linux Network Namespaces
- Software-defined networking (wikipedia)

Задание:

1. Найдите, какое сетевое решение используется в Minikube, какой блок адресов используется для Pods и какой - для Services
2. В ответе укажите при помощи каких команд вы это выполнили. Приведите команды и выводы в консоли.