??????? ???????????

Стартовая информация

Сетевой плагин выбирается во время установки кластера.

Документация, сетевая модель K8s Выдержки:

1. Сетевая подсистема используется только подами (не нодами).
2. У каждой node свой пул ip адресов для запущенных у них pod.
3. Каждый Pod в кластере имеет собственный уникальный в пределах кластера IP адрес.
4. У каждого Pod свое частное сетевое пространство, общее для всех контейнеров внутри Pod. Контейнеры внутри Pod взаимодействуют через localhost.
5. Pod'ы взаимодействуют с Pod на других нодах при помощи сетевого плагина (CNI).
6. Служба - это метод предоставления доступа к Pod в кластере (внутренний и внешний)

Вот вроде просто, но нихера не понятно. Поэтому дальше начинается самое веселое)

Следствия пункта 1. Связность между нодами и контроллером опосредованно зависит от сетевого взаимодействия подов. Поэтому вопрос сети при управлении - один вопрос, вопрос сети подов - второй вопрос.

Первый вопрос относительно простой: сетевая видимость точка - точка между IP адресами. Необходимые порты при настройке port-forwarding:

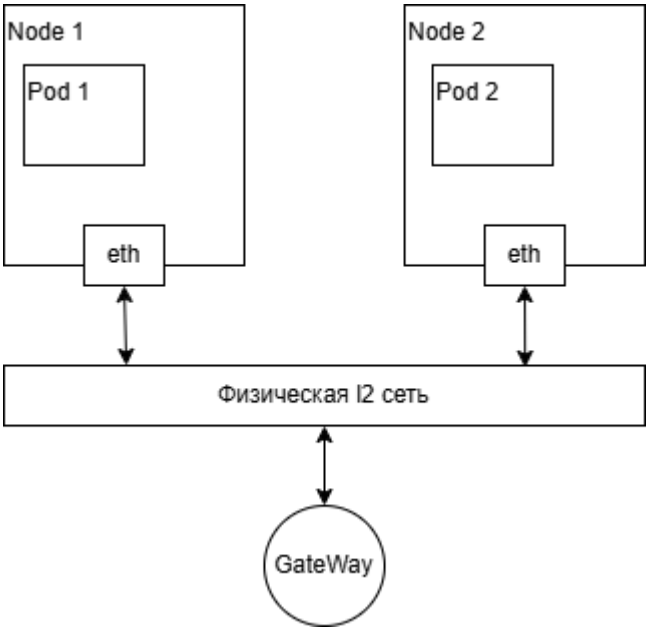
На мастер нодах:

```
TCP     6443*       Kubernetes API Server
TCP     2379-2380   etcd server client API
TCP     10250       Kubelet API
TCP     10251       kube-scheduler
TCP     10252       kube-controller-manager
TCP     10255       Read-Only Kubelet API
```

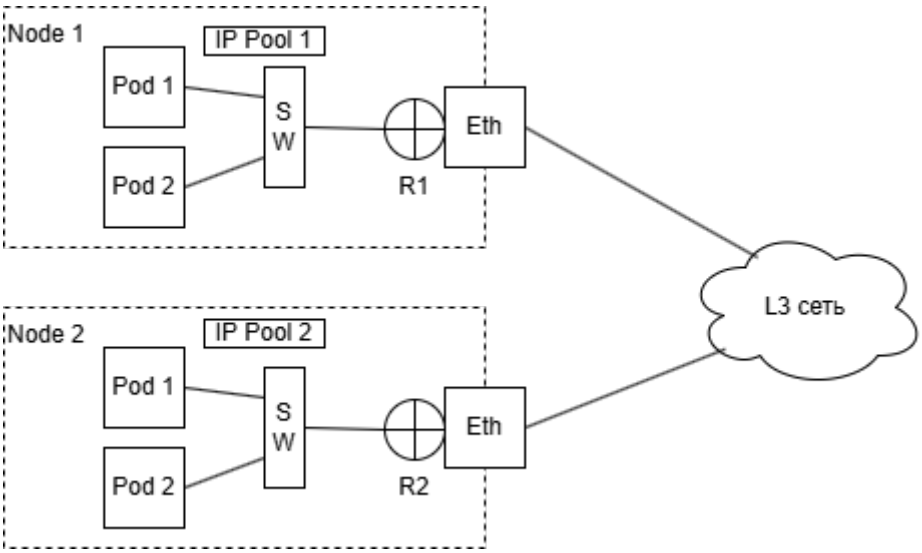
На воркерах:

```
TCP     10250       Kubelet API
TCP     10255       Read-Only Kubelet API
TCP     30000-32767 NodePort Services
```

Открываем порты и управление начнет работать. Теперь нужна сетевая связность между подами.



Для использования службы типа LoadBalancer необходимо установить балансировщик, например MetalLB. При существовании кластера в пределах одного L2 сегмента хватит L2 режима. Однако при разных сетях у worker потребуется BGP. Картинку можно представить следующим образом:



Роли маршрутизаторов R1 и R2 выполняют сетевые плагины (MetalLB, Calico, ...) в BGP режиме. Им необходим обмен маршрутной информацией в пределах L3 сети. Классическая сетевая задача - сделать видимым IP Pool 1 для IP Pool 2. Тут можно решить при помощи VXLAN и т д. В случае реализации без туннелей в L3 сети требуется, чтобы маршрутизаторы внутри этой сети также обладали маршрутной информацией.

Настройка MetalLB

Примените манифест для установки MetalLB

```
kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.13.7/config/manifests/metallb-native.yaml
```

Дождаться запуска контейнеров

```
kubectl get pods -n metallb-system
```

Создайте конфигурационный файл для MetalLB. Например, `metallb-config.yaml`

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: first-pool
  namespace: metallb-system
spec:
  addresses:
  - 192.168.1.240-192.168.1.250  # Укажите диапазон IP-адресов, доступных в вашей сети
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2advert
  namespace: metallb-system
spec:
  ipAddressPools:
  - first-pool
```

Применить конфигурацию:

```
kubectl apply -f metallb-config.yaml
```

Revision #12
Created 19 March 2025 16:30:12 by Admin
Updated 30 August 2025 15:36:12 by Admin