# K8s

## 服务发现

环境变量：podA里的应用需要访问serviceB里的服务，通过环境变量将serviceB的ip和port注入podA。这要求serviceB必须在podA之前被创建。

dns：kubeadm自带coredns。可以让podA通过serviceB的名称来访问serviceB，[service-name].[namespace].svc.cluster.local就是serviceB的域名。

## 服务发布

三种方式：

NodePort：在宿主机上创建一个端口，可以通过宿主机ip+NodePort的方式访问集群内部服务。

NodePort的范围是通过--service-node-port-range配置项来指定的，默认值为（30000-32767）。

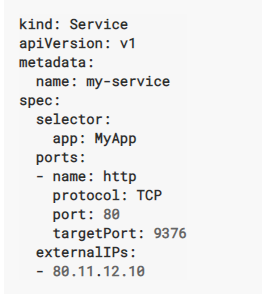
还可以通过kube-proxy的--nodeport-addresses选项来决定可以被用作NodePort绑定的ip，比如127.0.0.0/8这种。这个选项默认是[]，相当于NodePort会被绑定在0.0.0.0的ip上。

LoadBalancer：通过云厂商提供的LoadBalancer来访问集群内部服务。

ExternalName：通过服务名称来访问集群内部服务。

通过在yaml文件中配置externalName来给指定service的名称。

Externel IPs：如果有外部ip能够联通一个或者多个node节点的话，就可以使用这类ip作为service访问的入口。



比如这种就可以通过80.11.12.10:80来访问service。

## Ingress

Ingress提供了集群外部通过域名访问内部服务的能力，还提供了负载均衡。Ingress只针对HTTP和HTTPS协议有用，对于其他协议还是需要用到NodePort或者LoadBalancer。

使用Ingress之前需要安装Ingress controller。

Ingress本身也需要创建一个service来提供给外部集群来访问。根据http请求的host字段，找到对应的servicename和端口，然后将请求转发到对应的service里。

Ingress不要求应用容器一定要是按照deployment方式部署的，单个pod也是可行的。

Ingress搭建完成之后外部需要通过域名来访问还需要具备将域名转换成master节点ip的能力，单机测试可以通过修改host，网络的话就是通过dns。

Ingress需要跟service在相同的命名空间，不同的命名空间，ingress无法实现外部访问内部服务。

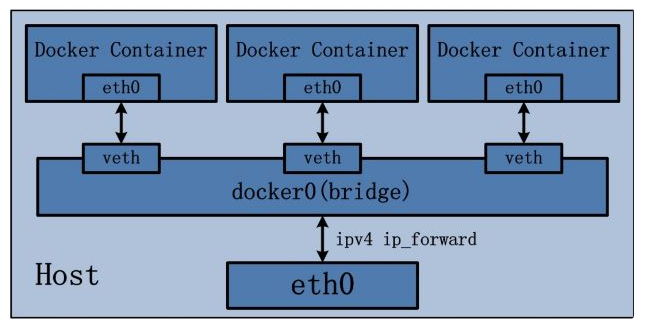
## Service的负载均衡

service本身是一个逻辑上的概念，是依赖各个node上的kube-proxy来将请求转发到service对应的各个pod上。请求转发到pod采用的是轮询的方式。可以通过service配置来实现请求定向转发到某一个pod上。

## 网络模型

Docker网络模型

Docker默认是采用brige网络模式，同时还提供了host、none、overlay等其他网络模式。

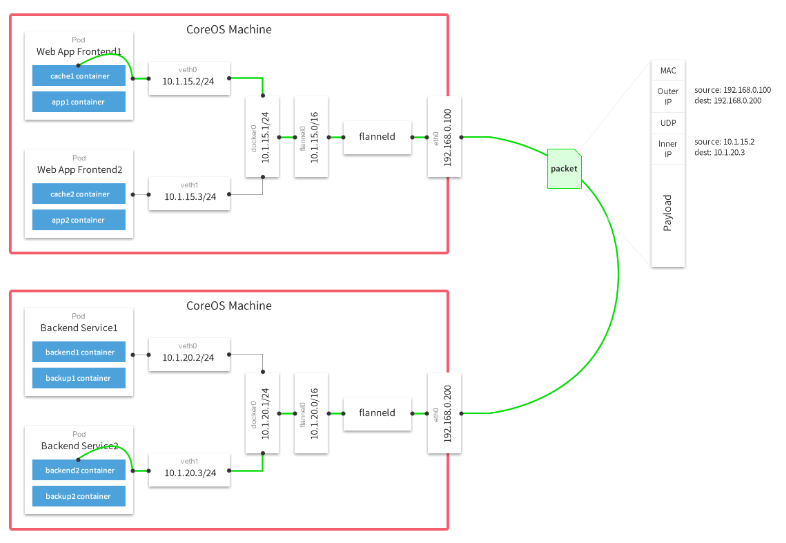


docker0网桥是在安装docker的时候就会生成。veth pair设备对是在启动容器的时候生成的。

一端在容器内部作为eth0网口，另一端在docker0上。作用就是将容器内发送到eth0网口的数据传递到docker0网桥上，再由网桥从宿主机的eth0接口发送出去。

docker0本身是无法被外部访问的，如果docker的容器需要被外部访问，就要绑定一个宿主机端口。

Flannel网络模型



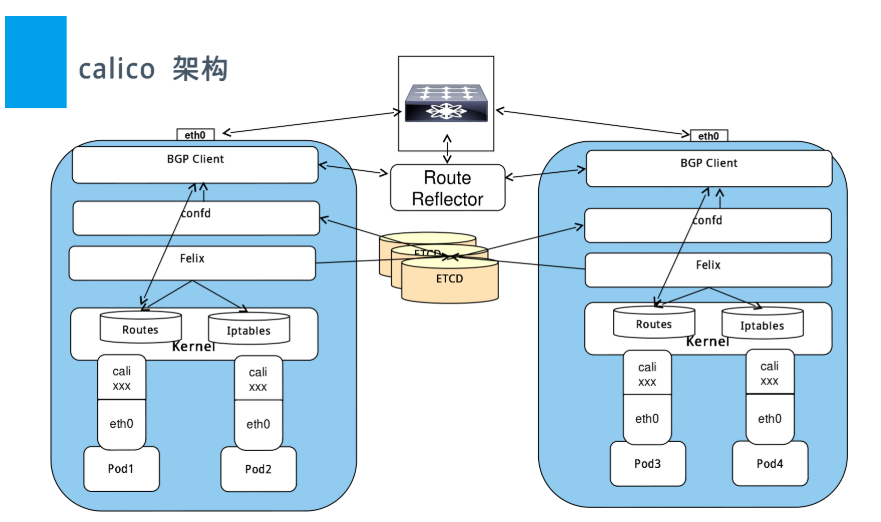
flannel网络插件为集群内每个节点上的docker分配不冲突的IP地址。同时为每个节点及docker创建一个overlay网络，实现数据包的互相传递。

flannel在master和node节点上启动了一个flanneld的服务。简单的说docker0网卡的数据原本是直接发给宿主机的eth0，现在发给flannel1网口，由flanneld服务对数据进行二次封装。flanneld会给数据封装一个二层头部，携带了源容器flannel1网卡的MAC地址和目的容器flannel1网卡的MAC地址。之后将数据交给宿主机eth0网口。目标机器收到数据之后由flanneld的服务解封装置后将真实数据交给对应容器。

同一台宿主机上的Pod之间通过docker0网桥来通信，Pod的ip和docker0网桥在同一个网段的。

不同宿主机上的Pod之间通过flannel通信。

calico网络模型



Felix：负责路由配置和ACLS规则配置的下发，运行在每个node节点上。

BGP Client：负责把Felix写入kernel的信息分发到当前的calico网络中，确保节点间通信有效性。

BGP Route Reflector：大规模部署使用的时候，通过一个或者多个BGP Route Reflector完成集中式路由部署和分发。

calico网络有两种方式：

1. IPIP

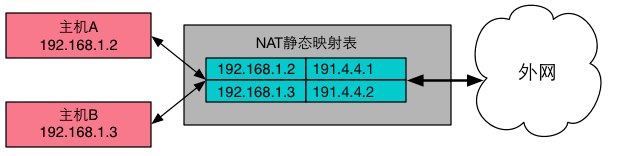
把一个ip数据包套在另一个ip报文中，相当于一个三层网桥。

1. BGP

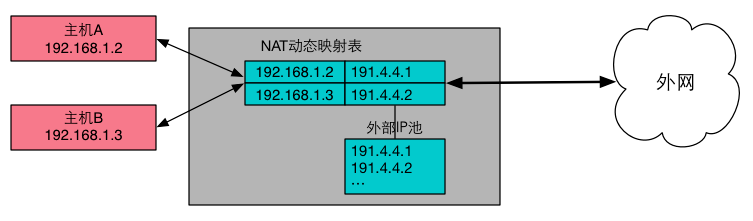
## NAT

NAT有三种模式：

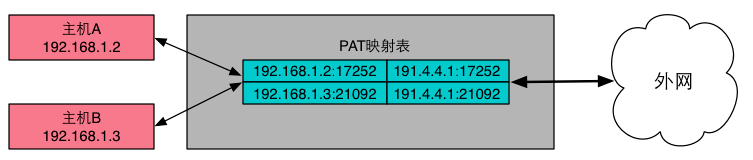
1. 静态NAT：将内部网络的私有IP地址转换为公有IP地址。私有地址和公有地址的对应关系是固定的。



1. 动态NAT：有内网IP需要转换为公网IP的时候，从公网IP池里取出一个IP，和内网IP形成对应关系。当通信结束之后，将公网IP返回IP池。



1. NAPT：端口层面的地址转换，对外使用统一的公网IP。根据不同的端口来做地址转换。



LVS