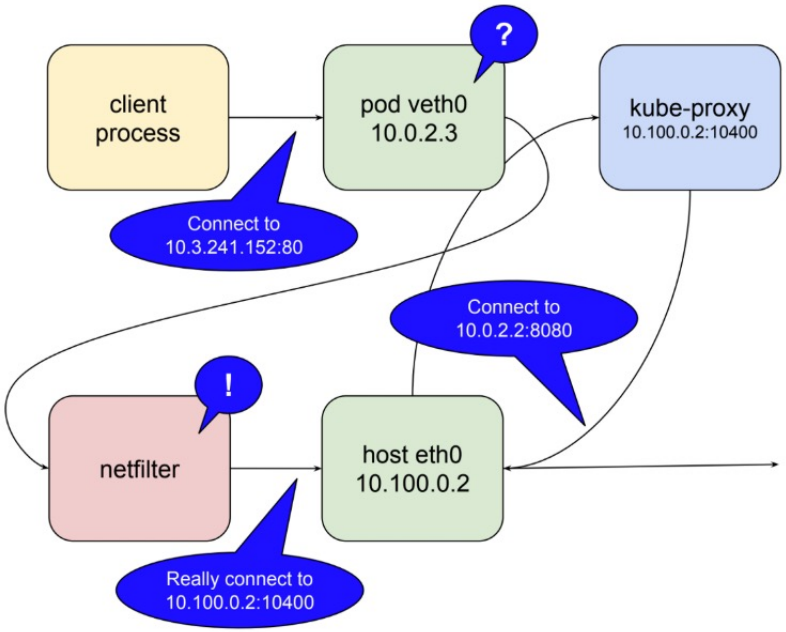
[Network] Services & Ingress

Services 개요

- 여러 레플리카에 트래픽을 분산시키는 로드밸런서 (TCP, UDP 모두 가능)
- 노드의 kube-proxy 를 활용하여 엔드포인트로 라우팅 되도록 처리

[Kube-proxy가 직접 UserSpace Proxy역할 시]



[Kube-proxy가 Iptables를 통해 netfilter조작하는 역할 시]

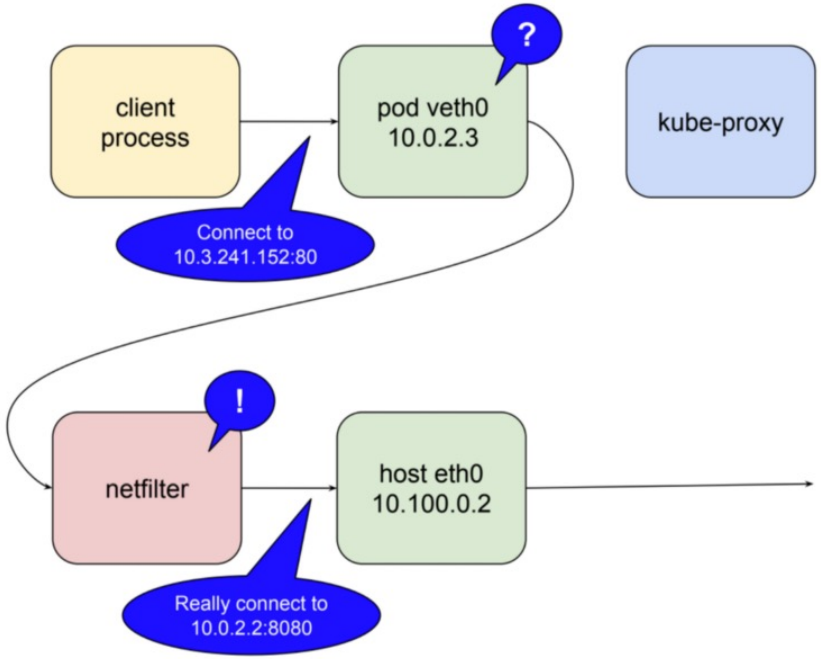


그림 출처 : https://medium.com/finda-tech/kubernetes-네트워크-정리-fccd4fd0ae6

Services 실습

- https://kubernetes.io/ko/docs/concepts/services-networking/connect-applications-service 참고

```
[centos@osk-master-01 ~]$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/website/main/content/ko/
examples/service/networking/run-my-nginx.yaml
deployment.apps/my-nginx created
[centos@osk-master-01 ~]$ kubectl expose deployment/my-nginx
service/my-nginx exposed
[centos@osk-master-01 ~]$ kubectl describe svc my-nginx
Name:              my-nginx
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          run=my-nginx
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.104.164.89
IPs:               10.104.164.89
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         192.168.183.140:80,192.168.224.7:80
Session Affinity:  None
Events:            <none>
[centos@osk-master-01 ~]$
```

DNS

- (위키백과) 도메인 네임 시스템(Domain Name System, DNS)은 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있도록 하기 위해 개발되었다. 특정 컴퓨터(또는 네트워크로 연결된 임의의 장치)의 주소를 찾기 위해, 사람이 이해하기 쉬운 도메인 이름을 숫자로 된 식별 번호(IP 주소)로 변환해 준다. 도메인 네임 시스템은 흔히 "전화번호부"에 비유된다. 인터넷 도메인 주소 체계로서 TCP/IP의 응용에서, www.example.com과 같은 주 컴퓨터의 도메인 이름을 192.168.1.0과 같은 IP 주소로 변환하고 라우팅 정보를 제공하는 분산형 데이터베이스 시스템이다.

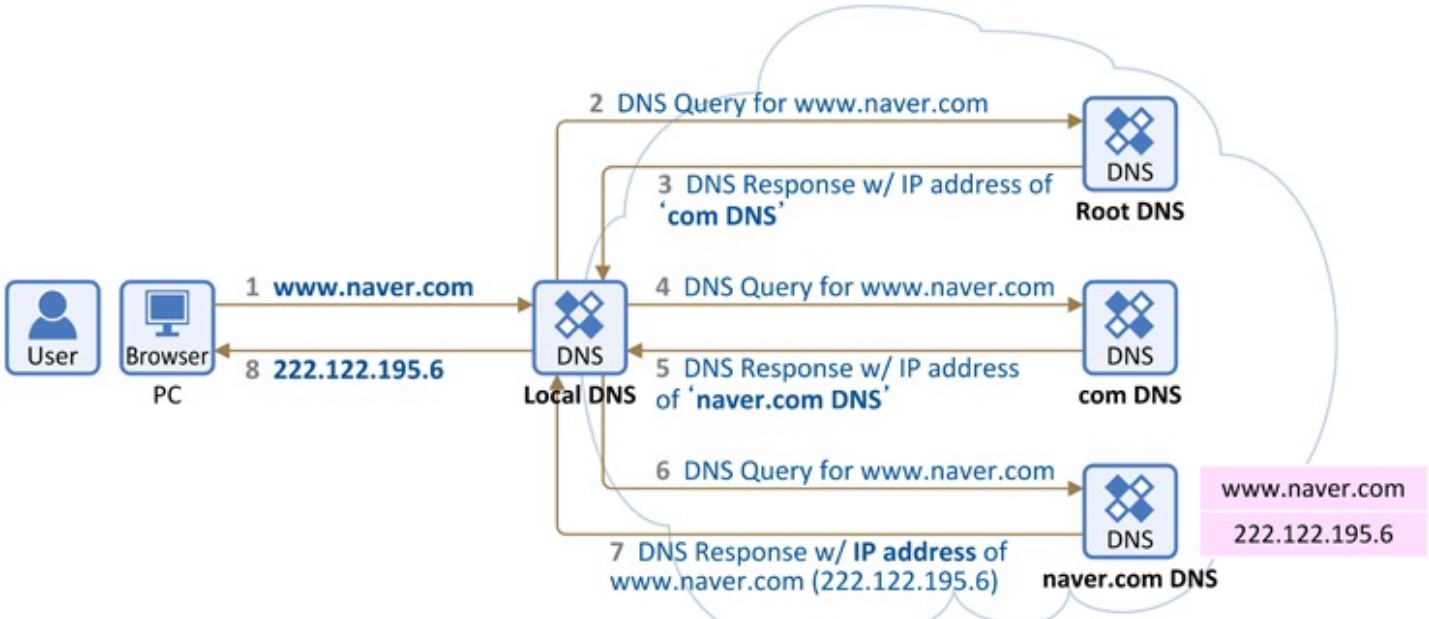


그림 출처 : https://www.netmanias.com/ko/post/blog/5353/dns/dns-basic-operation

DNS

- 공유기에서는 단말에 사설 IP를 할당하면서 DNS 주소도 배포 가능

- 공유기, DNS 를 모두 나쁜 목적으로 활용 시 :

그림 출처 : https://blog.skbroadband.com/2752

DNS

- 설치 과정 중, 기본 값인 CoreDNS를 애드온으로 설치 완료

```
[centos@osk-master-01 ~]$ kubectl get svc -A
NAMESPACE     NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP                  44h
default       my-nginx     ClusterIP   10.104.164.89   <none>        80/TCP                   20m
kube-system   kube-dns     ClusterIP   10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP   44h

[centos@osk-master-01 ~]$ kubectl run --image=alpine dns-test -it -- /bin/sh
If you don't see a command prompt, try pressing enter.
/ # nslookup kubernetes
Name: kubernetes.default.svc.cluster.local  # 모든 서비스는 생성시 <service name>.<namespace>.svc.cluster.local
Address: 10.96.0.1
생략
/ # nslookup my-nginx
Name: my-nginx.default.svc.cluster.local
Address: 10.104.164.89
/ # cat /etc/resolv.conf
nameserver 10.96.0.10
search default.svc.cluster.local svc.cluster.local cluster.local kr-central-1.c.kakaoi.io kr-central-1.c.internal
options ndots:5
```

Ingress

- 클러스터 내의 서비스에 대한 외부 접근을 관리하는 API 오브젝트이며, 일반적으로 HTTP를 관리함.
- (시나리오) 앞서 Service 과정에서 만든 my-nginx는 ClusterIP 서비스이기에 클러스터 내에서는 접속 가능하지만 외부에서는 접속 불가

```
[centos@osk-master-01 ~]$ kubectl describe svc my-nginx
Name:                   my-nginx
Type:                   ClusterIP
```

- 외부에 load balancing을 지원하는 Ingress를 통해 외부에서도 nginx 서비스에 접속 가능토록 실습

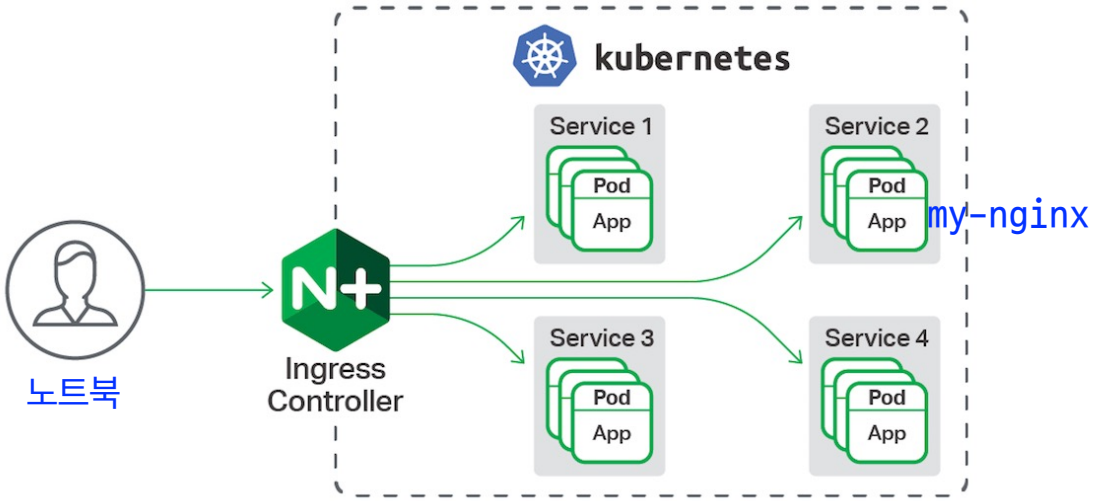


그림 출처 : https://www.nginx.com/blog/wait-which-nginx-ingress-controller-kubernetes-am-i-using/

Ingress

- https://kubernetes.io/ko/docs/concepts/services-networking/ingress/#인그레스-리소스 참고

```
[centos@osk-master-01 ~]$ wget https://raw.githubusercontent.com/kubernetes/website/main/content/ko/examples/service/networki
ng/minimal-ingress.yaml

[centos@osk-master-01 ~]$ cat minimal-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: my-nginx          → vi로 수정
            port:
              number: 80
```

Ingress Controller

- 앞서 생성한 Ingress는 로드밸런싱에 필요한 정보 일뿐, 실제 로드 밸런싱을 수행할 프로그램이 있어야함(인그레스 컨트롤러)
- 종류는 다양하며, 쿠버네티스 프로젝트에서는 AWS/GCE/nginx 지원 https://kubernetes.io/ko/docs/concepts/services-networking/ingress-controllers/

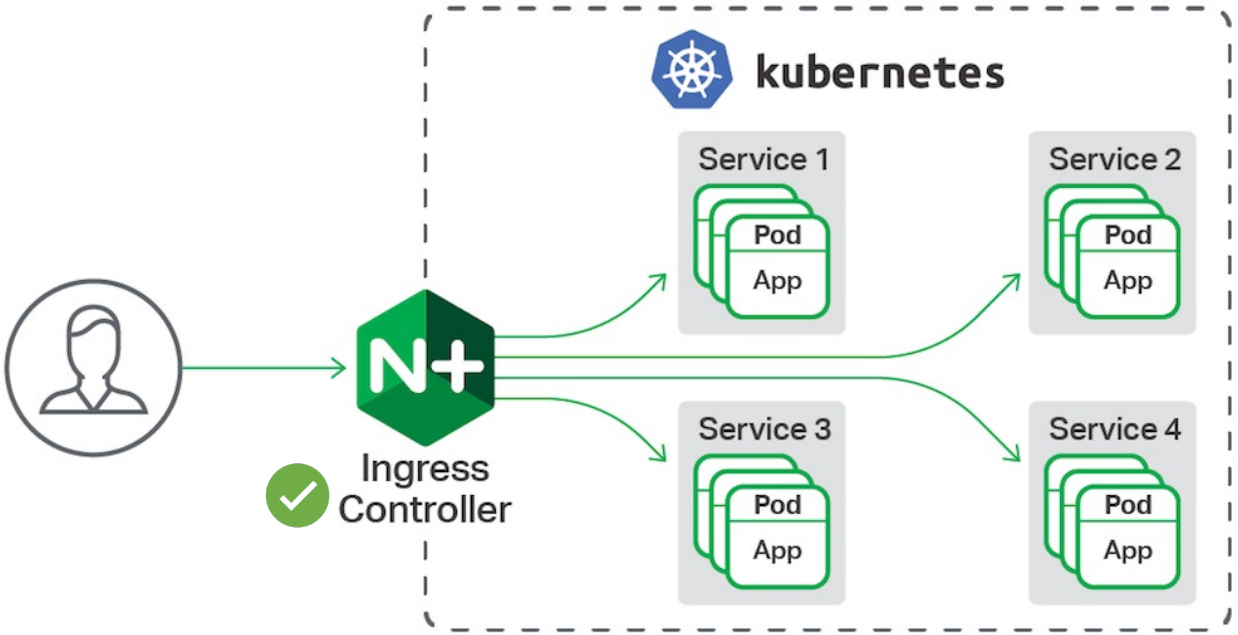


그림 출처 : https://www.nginx.com/blog/wait-which-nginx-ingress-controller-kubernetes-am-i-using/

Ingress Controller

- Ingress Controller 생성 https://kubernetes.github.io/ingress-nginx/deploy/#bare-metal

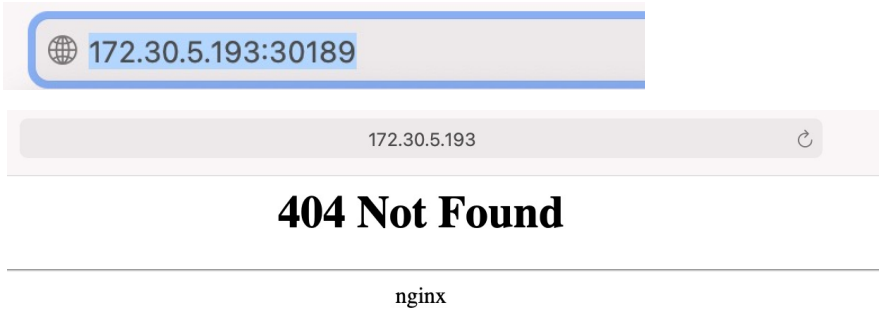
```
[centos@osk-master-01 ~]$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.48.1/deploy/static/provider/baremetal/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
configmap/ingress-nginx-controller created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
service/ingress-nginx-controller-admission created
service/ingress-nginx-controller created
deployment.apps/ingress-nginx-controller created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
serviceaccount/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
job.batch/ingress-nginx-admission-create created
job.batch/ingress-nginx-admission-patch created
```

Ingress Controller

- Ingress 생성

```
[centos@osk-master-01 ~]$ kubectl apply -f minimal-ingress.yaml
ingress.networking.k8s.io/minimal-ingress created
```

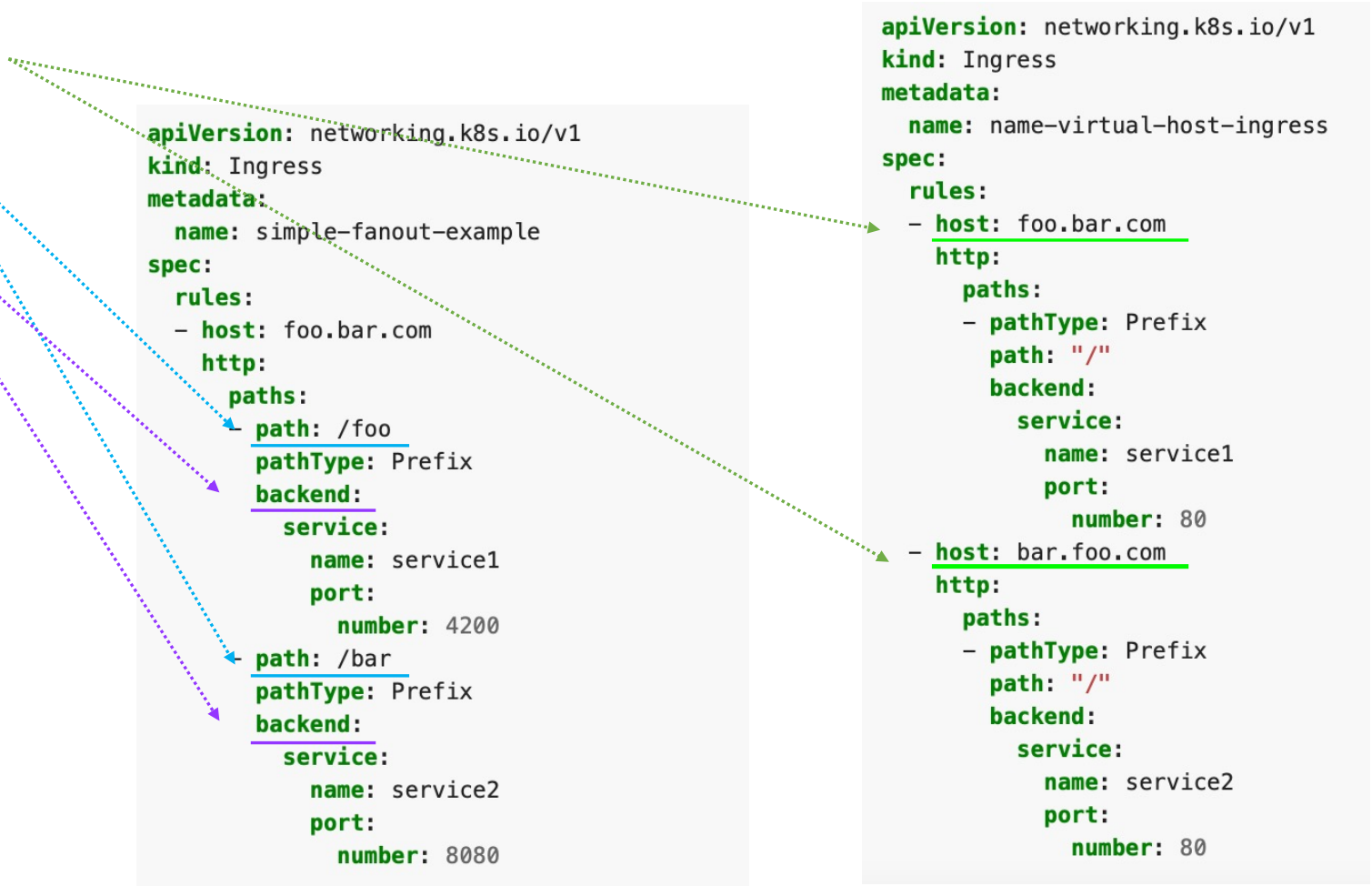
```
[centos@osk-master-01 ~]$ kubectl get service -n ingress-nginx
NAME                                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
ingress-nginx-controller             NodePort    10.109.83.45    <none>        80:30189/TCP,443:32660/TCP   3h8m
ingress-nginx-controller-admission   ClusterIP   10.98.63.210    <none>        443/TCP                      3h8m
```



- Ingress Class 관련 : https://kubernetes.github.io/ingress-nginx/#i-have-only-one-instance-of-the-ingresss-nginx-controller-in-my-cluster-what-should-i-do

Ingress Rule

- 1. 선택적 호스트
- 2. 경로목록
- 3. 백엔드



```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: simple-fanout-example
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        pathType: Prefix
        backend:
          service:
            name: service1
            port:
              number: 4200
      - path: /bar
        pathType: Prefix
        backend:
          service:
            name: service2
            port:
              number: 8080
```

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: name-virtual-host-ingress
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: service1
            port:
              number: 80
  - host: bar.foo.com
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: service2
            port:
              number: 80
```