



Expose service

By Minh Monmen



Nội dung

- Sơ lược về networking trong K8S
- Service: ClusterIP, NodePort, LoadBalancer
- Service discovery, Service DNS
- Ingress
- Kiến trúc expose ứng dụng thường dùng
- Thực hành 1: Triển khai ứng dụng đầy đủ trên K8S
- Thực hành 2: Expose nhiều ứng dụng qua ingress

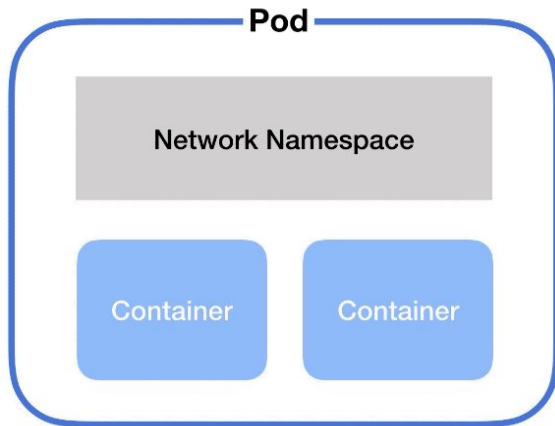


Sơ lược về networking trong K8S

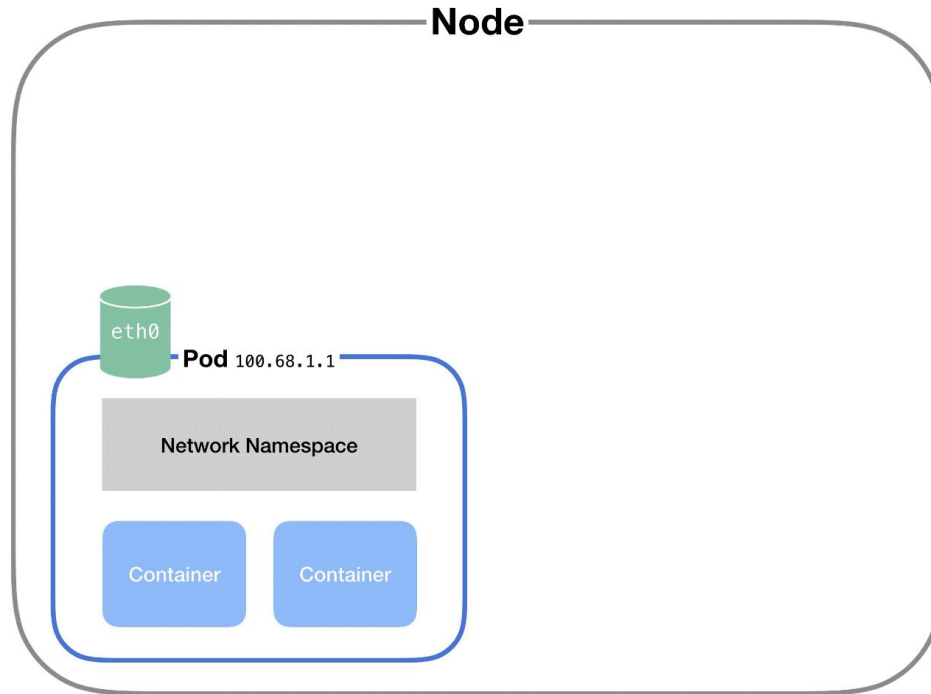
- Mỗi pod sẽ được cấp 1 địa chỉ IP nội bộ
- 1 pod có khả năng truy cập mọi IP pod khác trong cluster
- Các agent kube-proxy, kubelet có thể truy cập được mọi pod trên host đó



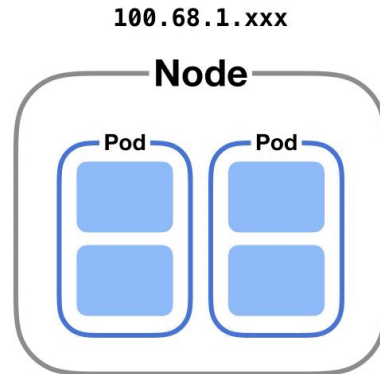
Network trong pod



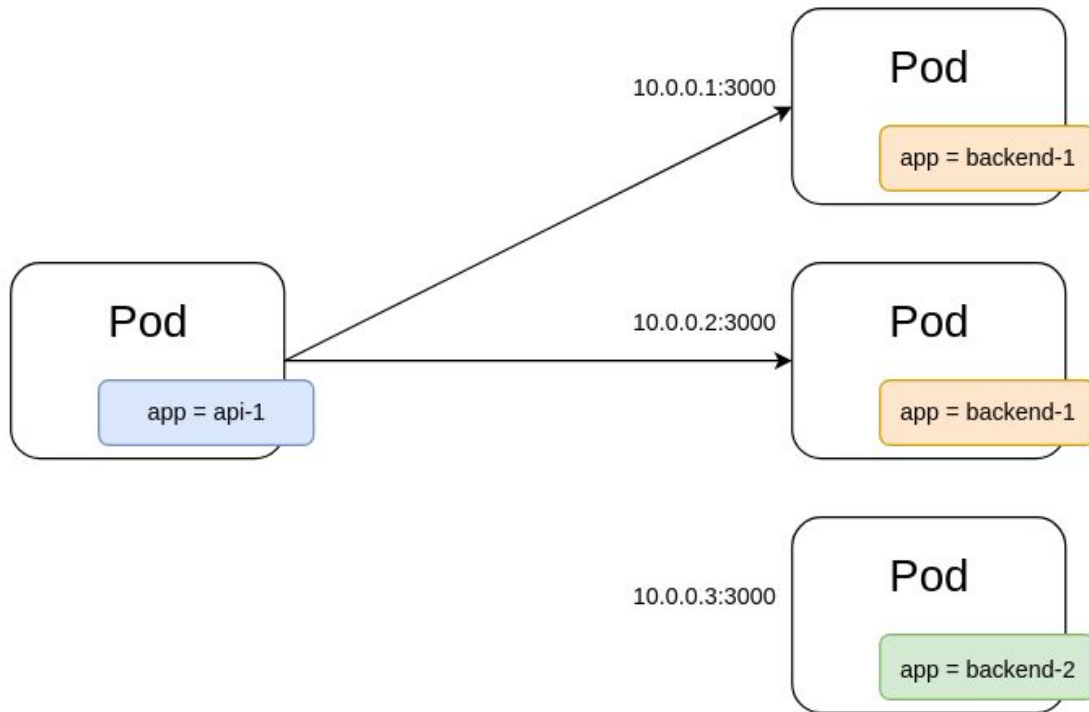
Network giữa các pod trong 1 host



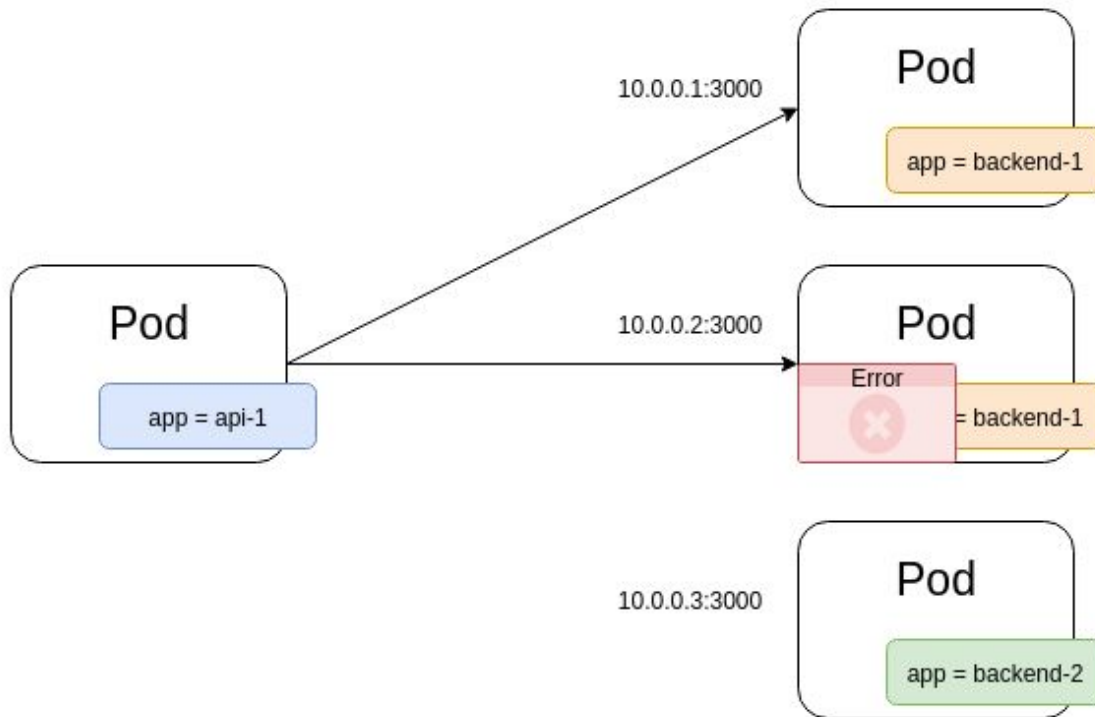
Network giữa các pod trên nhiều host



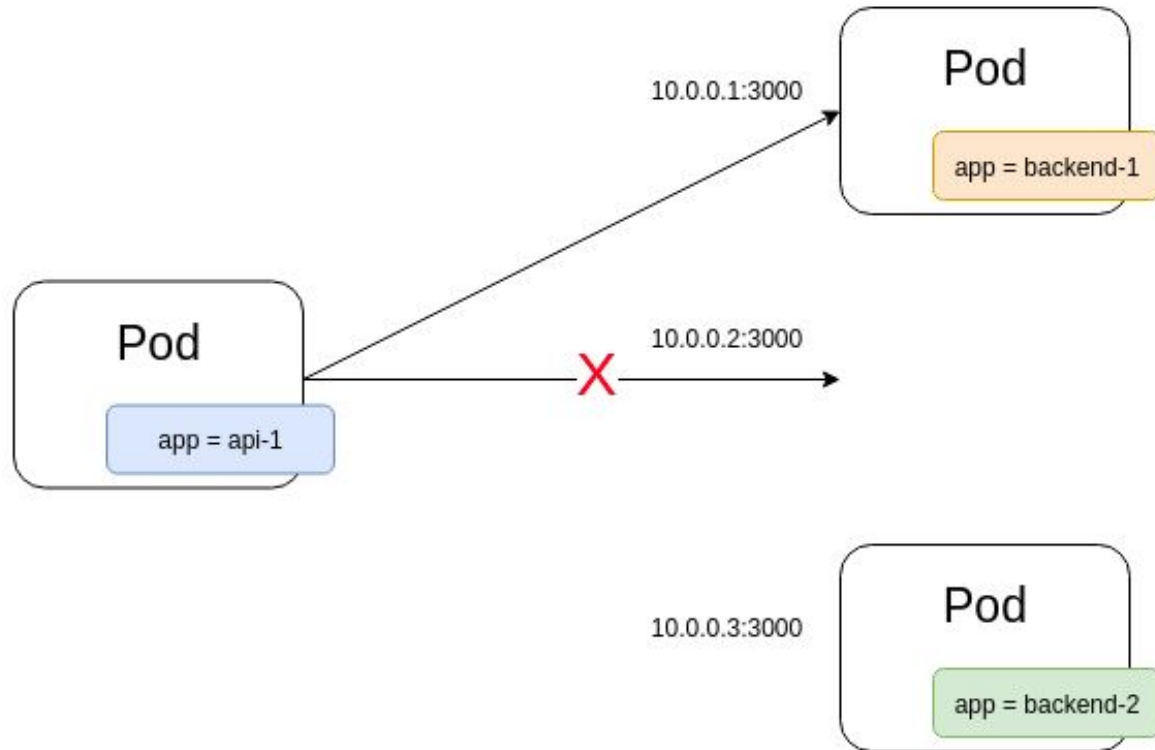
Sự cần thiết của service



Sự cần thiết của service



Sự cần thiết của service

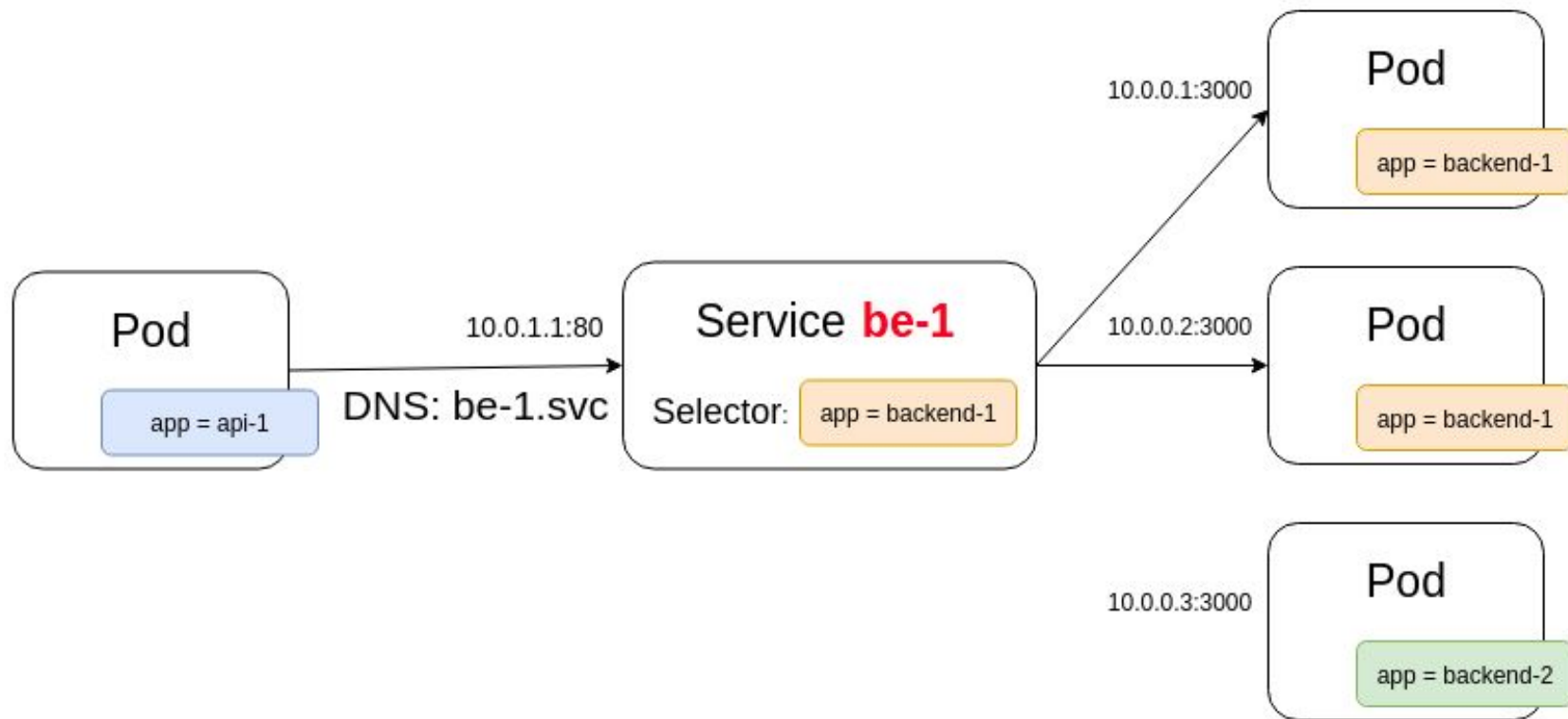




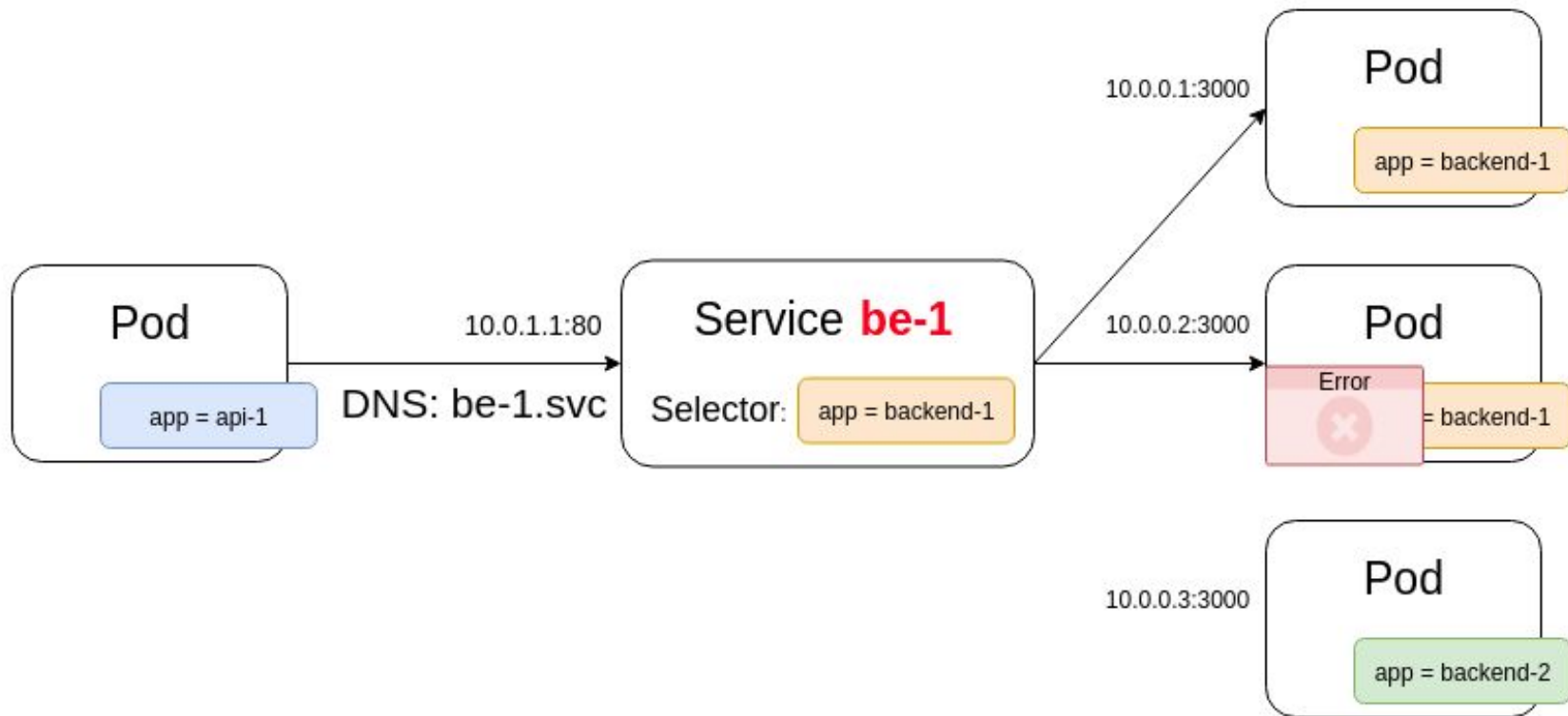
Service kèm proxy

- Mỗi pod sẽ có 1 IP riêng
- **Tập hợp các pod bằng label qua 1 Virtual IP chung**
- Proxy bằng iptables hoặc ipvs
- Auto load balancing (round-robin)
- Auto discovery khi pod có label được khởi chạy hoặc dừng

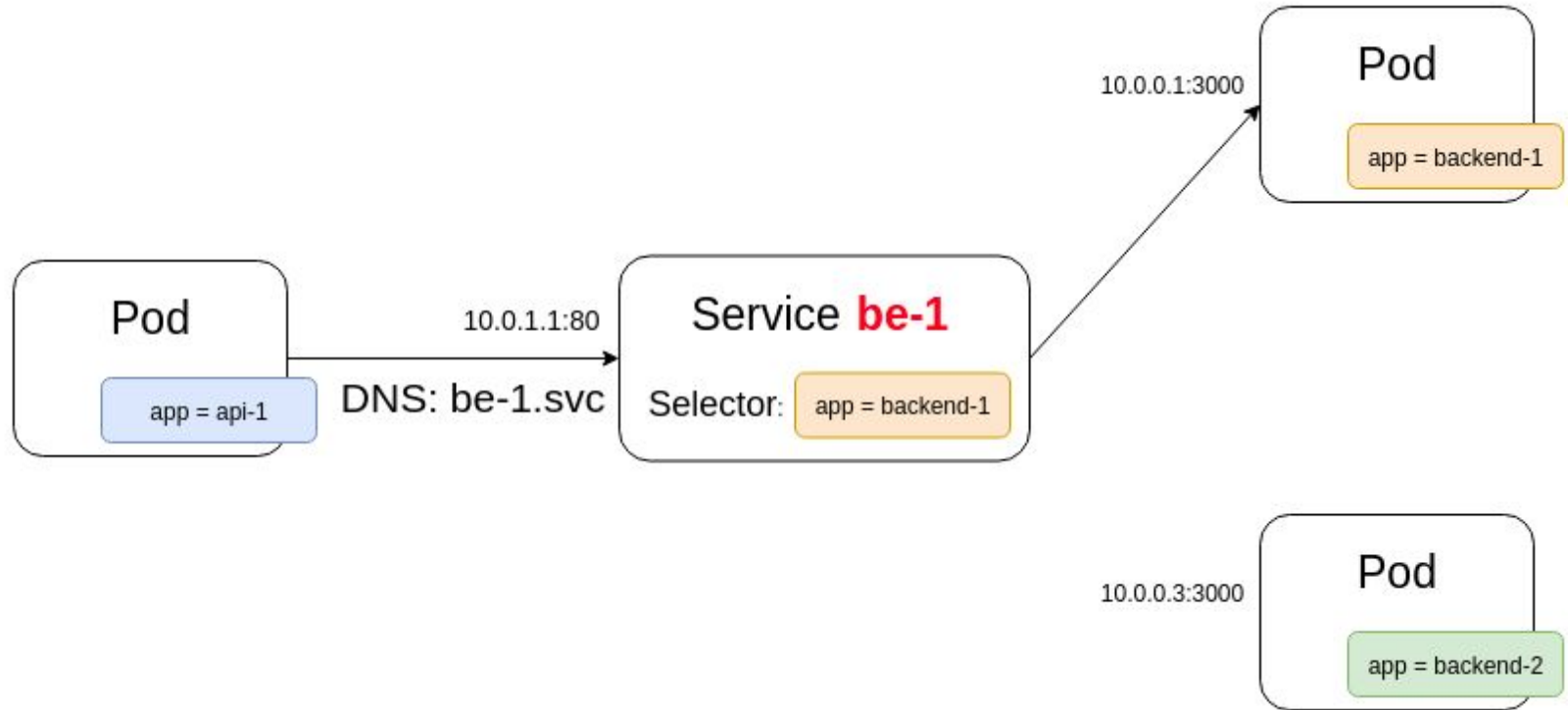
Service kèm proxy



Service kèm proxy

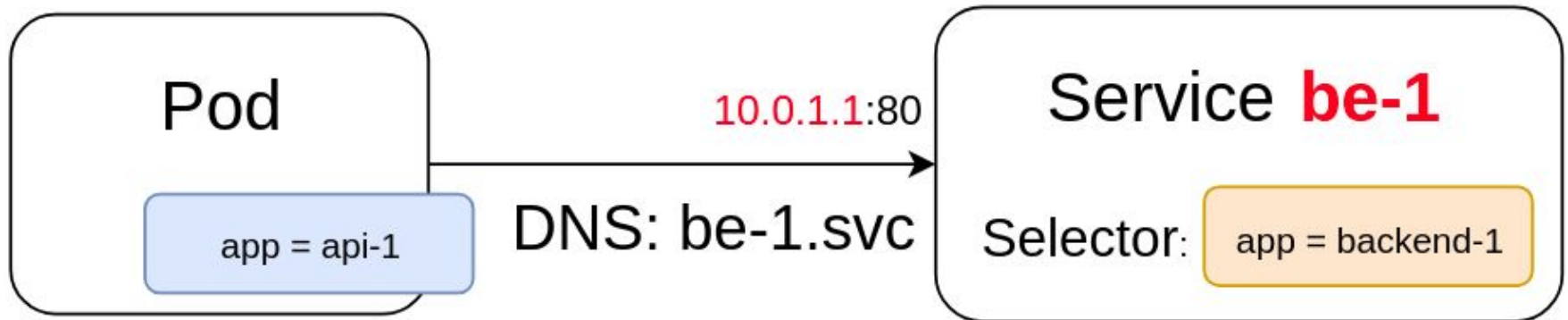


Service kèm proxy



Service kèm proxy: ClusterIP

- Tạo ra 1 Virtual IP trong cluster
- Dùng cho các trường hợp gọi giữa các pod với nhau



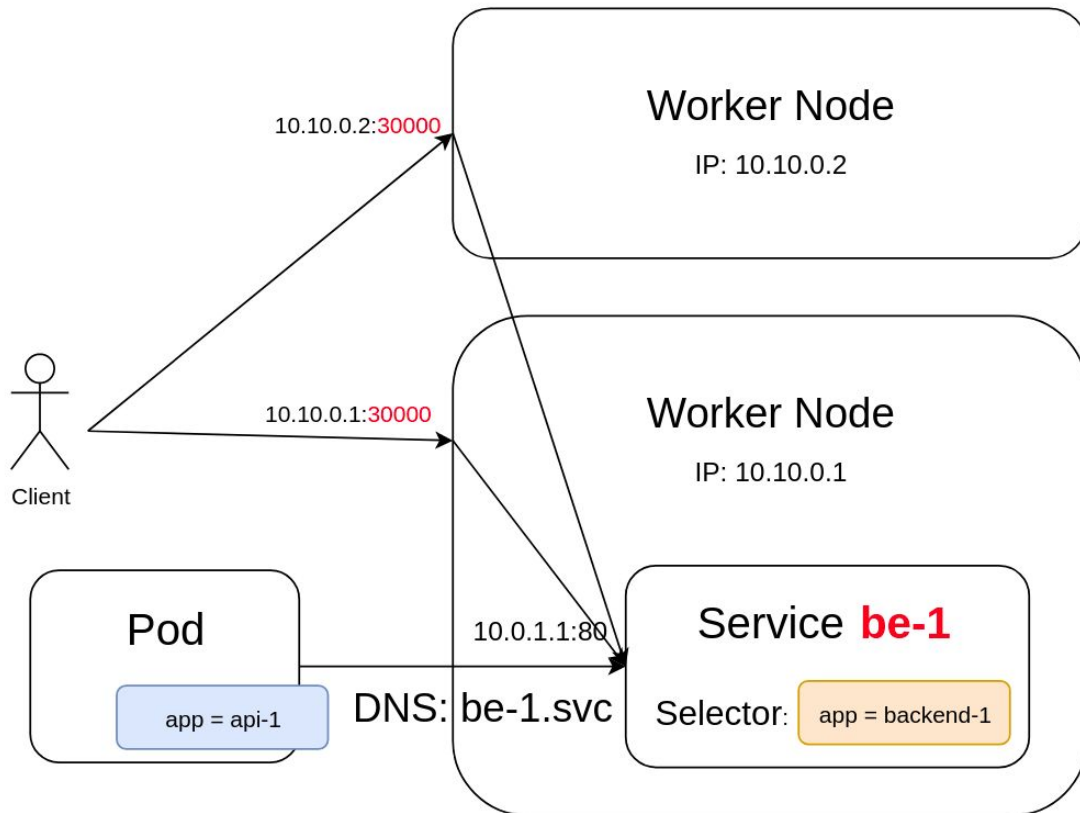


ClusterIP service

```
apiVersion: v1
kind: Service
metadata:
  name: demo-service
spec:
  type: ClusterIP          # Chỉ tạo Virtual IP
  selector:
    app: backend-1        # Label selector
  ports:
    - protocol: TCP        # Protocol
      port: 80             # Port của service
      targetPort: 3000     # Port của pod
```

Service kèm proxy: NodePort

- Tạo ClusterIP
- Map port trên MỌI host tới service port
- Dùng để truy cập service từ bên ngoài
- Port range: 30000~32767



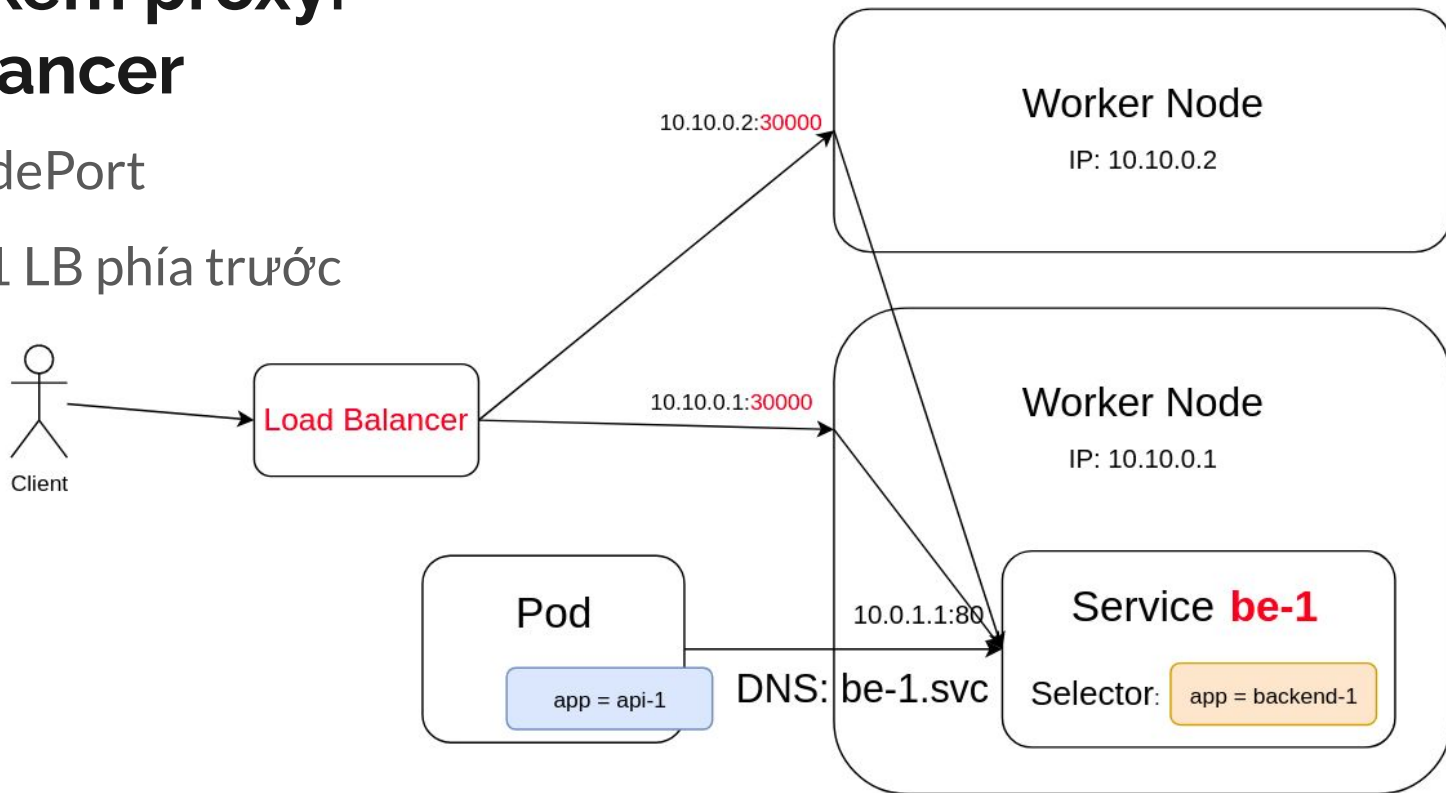


NodePort Service

```
---
apiVersion: v1
kind: Service
metadata:
  name: demo-service
spec:
  type: NodePort          # Virtual IP + map host port
  selector:
    app: backend-1        # Label selector
  ports:
    - protocol: TCP        # Protocol
      port: 80             # Port của service
      targetPort: 3000      # Port của pod
      nodePort: 30007       # Port của host (optional)
```

Service kèm proxy: LoadBalancer

- Tạo NodePort
- Tự tạo 1 LB phía trước





LoadBalancer Service

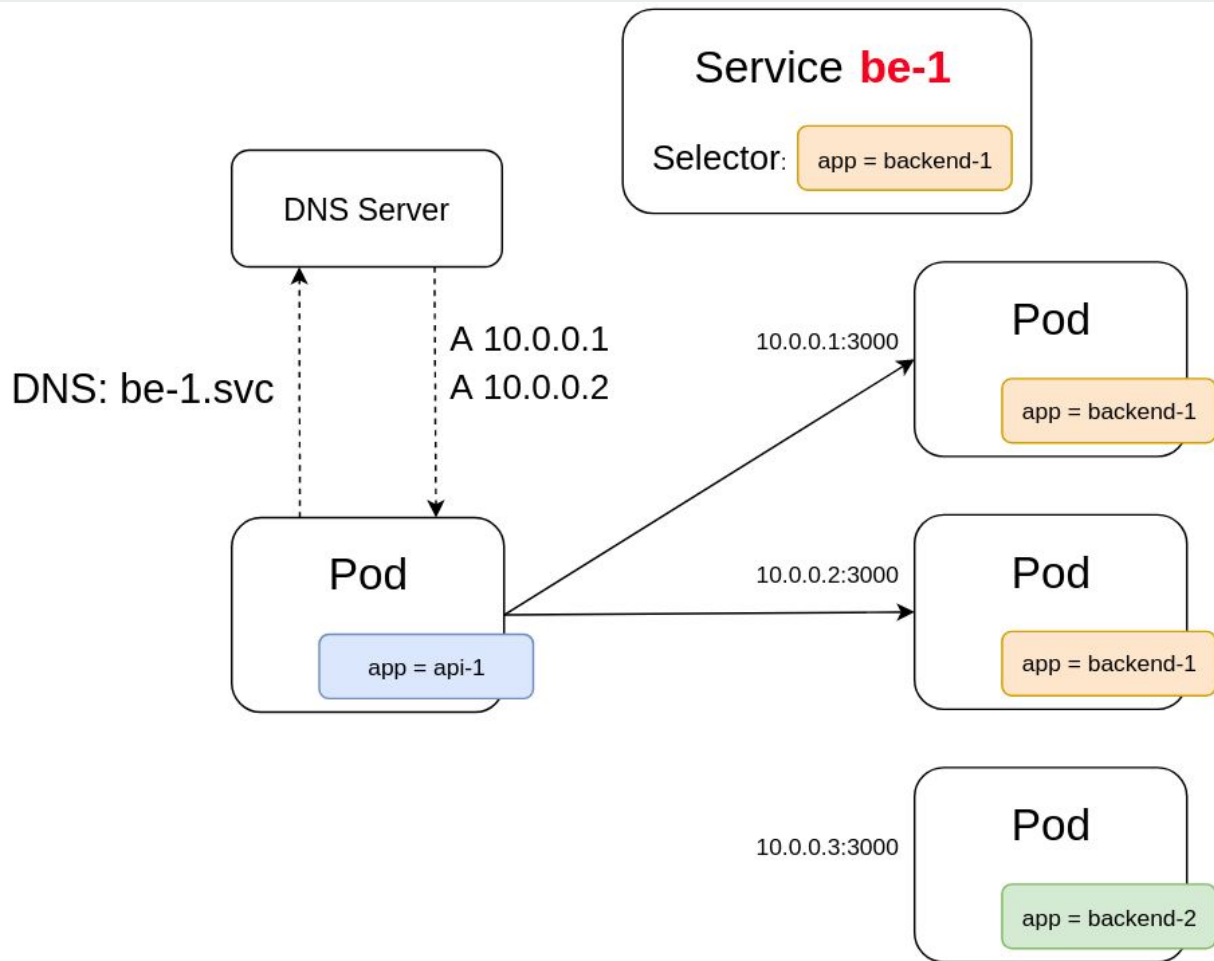
```
---
apiVersion: v1
kind: Service
metadata:
  name: demo-service
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-backend-protocol: http
    service.beta.kubernetes.io/aws-load-balancer-ssl-ports: "443,8443"
spec:
  type: LoadBalancer      # Virtual IP + map host port + tạo LB
  selector:
    app: backend-1         # Label selector
  ports:
    - protocol: TCP        # Protocol
      port: 80             # Port của service
      targetPort: 3000      # Port của pod
      nodePort: 30007       # Port của host (optional)
```



Service KHÔNG proxy (Headless)

- Phân giải DNS CNAME và A
- Dựa vào cơ chế DNS load balancing của client
- Dùng với trường hợp không cần 1 IP duy nhất
- Dùng với các service bên ngoài

Service KHÔNG proxy





Headless Service

```
---
apiVersion: v1
kind: Service
metadata:
  name: demo-service
spec:
  clusterIP: None           # Không tạo Virtual IP
  selector:
    app: backend-1         # Label selector
  ports:
    - protocol: TCP        # Protocol
      port: 80             # Port của service
      targetPort: 3000     # Port của pod
```



Service DNS

- `<service-name>`
- `<service-name>.<ns>`
- `<service-name>.<ns>.svc`
- `<service-name>.<ns>.svc.cluster.local.`
- `<service-name>.<ns>.svc.<cluster-domain>`



Thực hành 1: Deploy ứng dụng CRUD

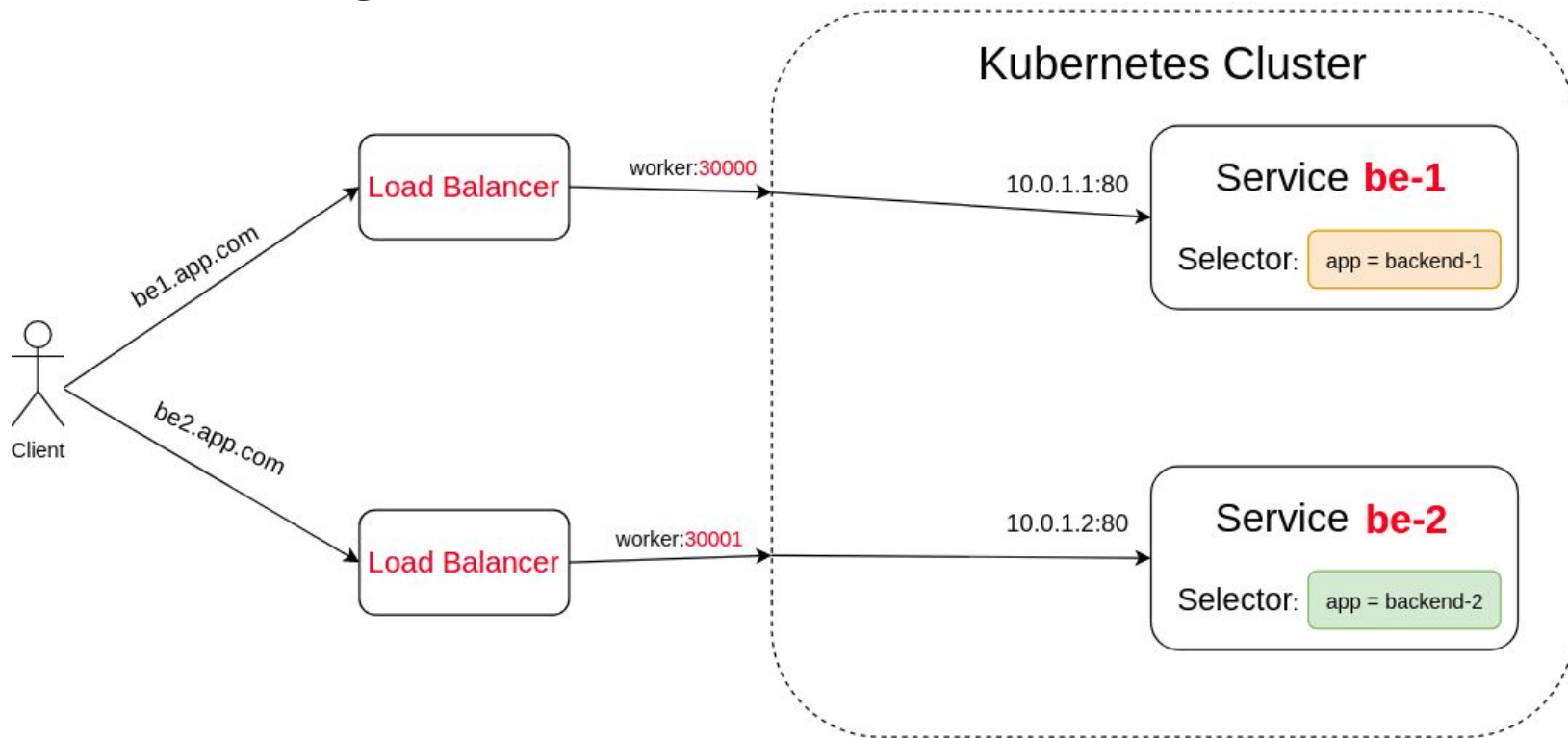
- <https://github.com/minhpg331/demo-service>
- Deploy ứng dụng CRUD bằng NodeJS và MongoDB trên K8S
- Task 1: Expose service mongodb cho app NodeJS sử dụng
- Task 2: Expose service NodeJS để truy cập từ bên ngoài
- Các yêu cầu và các bước làm có trong readme



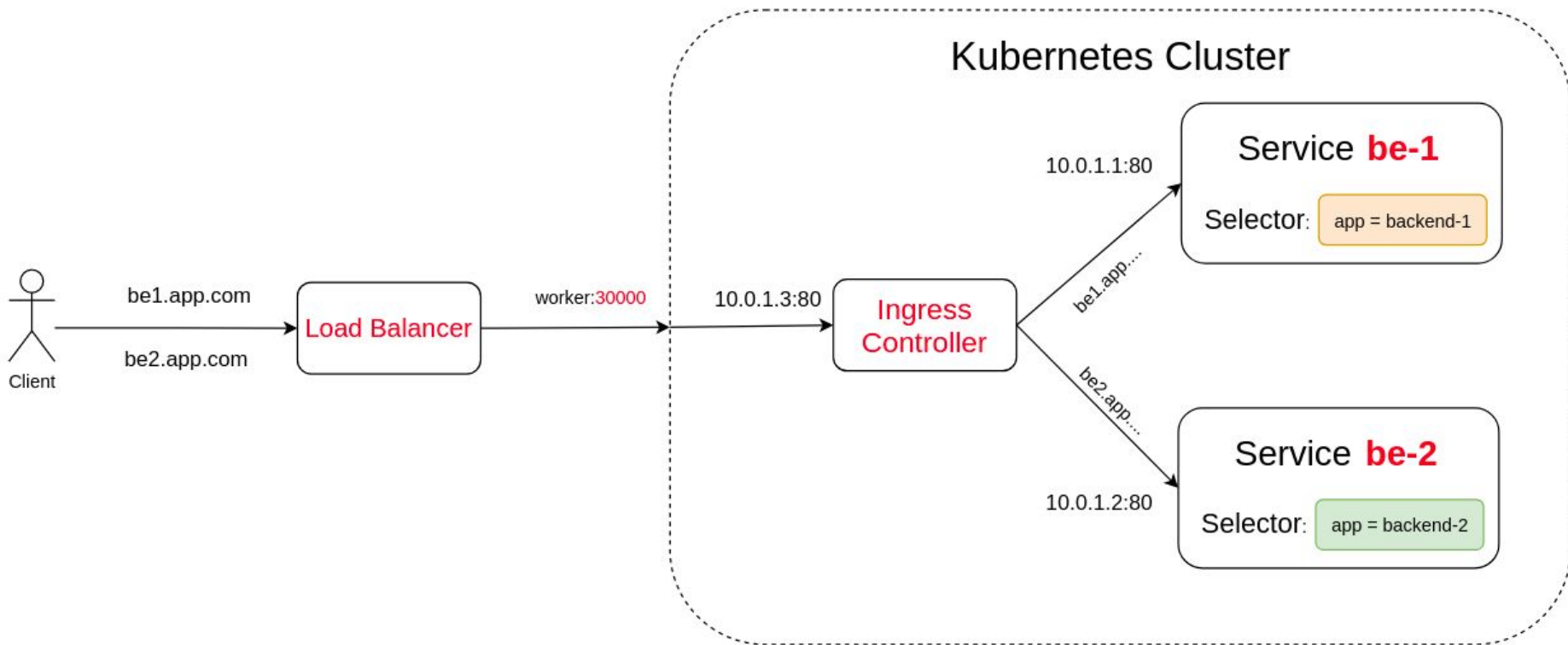
Ingress

- Mỗi service lại cần mở 1 node port hoặc cần 1 LB
- Expose duy nhất 1 service là Ingress Controller (như nginx)
- Routing qua Ingress Resource (như vhost block)
- Được quản lý bởi K8S
- Update routing tự động khi thay đổi ingress

Without Ingress



Ingress





Cài đặt Ingress Controller

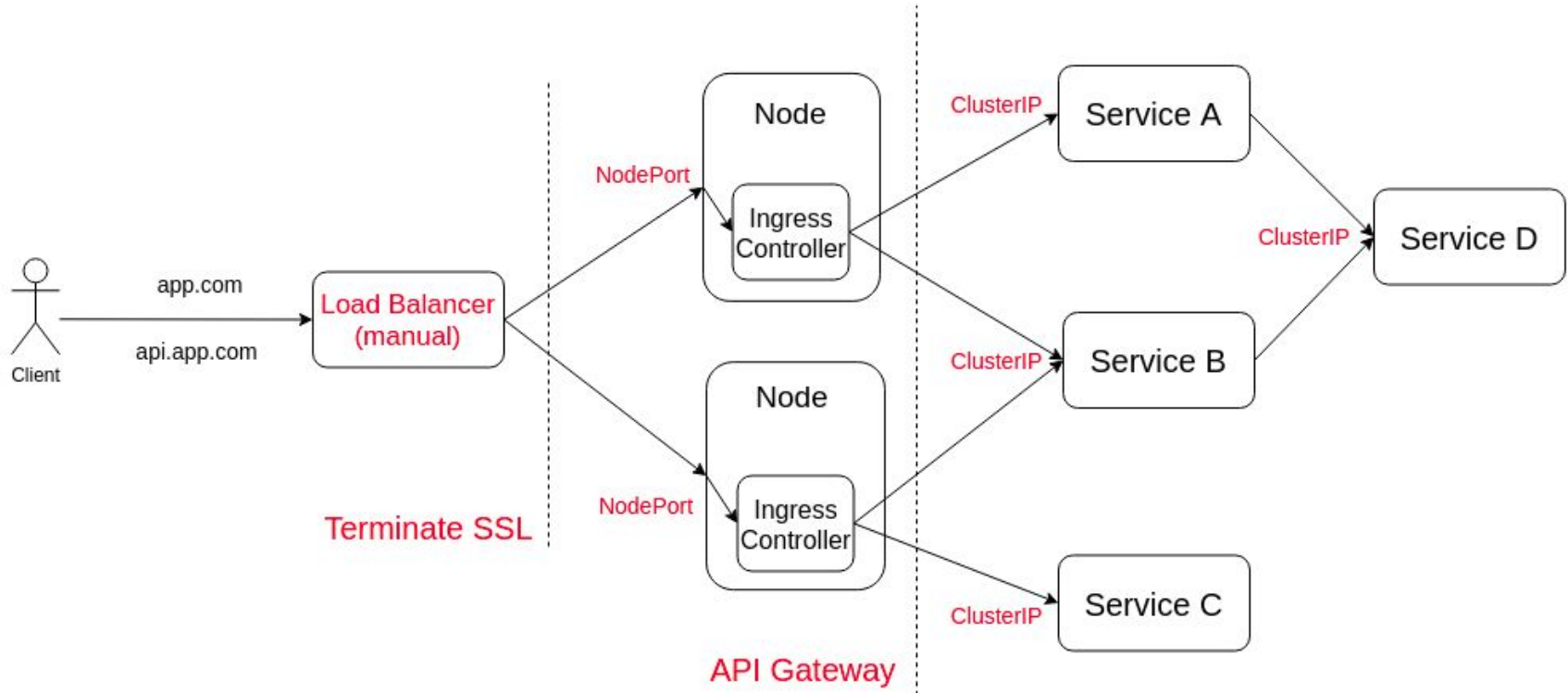
- So sánh ingress controller: [Kubernetes Ingress controllers comparison \(upd'2020\)](#)
- Cài đặt Nginx Ingress Controller (chỉ chạy 1 lần)
 - `kubectl apply -f`
`https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.35.0/deploy/static/provider/baremetal/deploy.yaml`



Định nghĩa Ingress

```
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: demo-ingress
spec:
  rules:
  - host: "app.com"                                # Domain name
    http:
      paths:
      - pathType: Prefix                            # Loại path
        path: "/"                                   # Path
        backend:
          service:
            name: frontend                          # Tên service đích
            port:
              number: 80
  - host: "api.app.com"
    http:
      paths:
      - pathType: Prefix
        path: "/api"
        backend:
          service:
            name: backend-1
            port:
              number: 80
```

Kiến trúc expose ứng dụng thường dùng





Thực hành 2: Expose service bằng ingress

- <https://github.com/minhpg331/demo-service>
- Task 3: Sử dụng nginx ingress controller để truy cập ứng dụng



1 số kinh nghiệm triển khai service

- Luôn dùng full DNS khi gọi tên service (giúp phân giải nhanh và chuẩn xác)
- Scale DNS service tới 1 mức phù hợp (rất hay quá tải)
- Để ý cơ chế phân giải DNS của app (ví dụ PHP mỗi kết nối DB lại phải phân giải lại ~> không nên dùng headless mà dùng IP)
- Fix nodePort (mặc định sẽ random)
- Config terminate SSL chuẩn xác (không sẽ mất IP nguồn)



Tài liệu tham khảo

- <https://matthewpalmer.net/kubernetes-app-developer/articles/kubernetes-networking-guide-beginners.html>
- <https://kubernetes.io/docs/concepts/cluster-administration/networking/>
- <https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>



Thanks for watching!