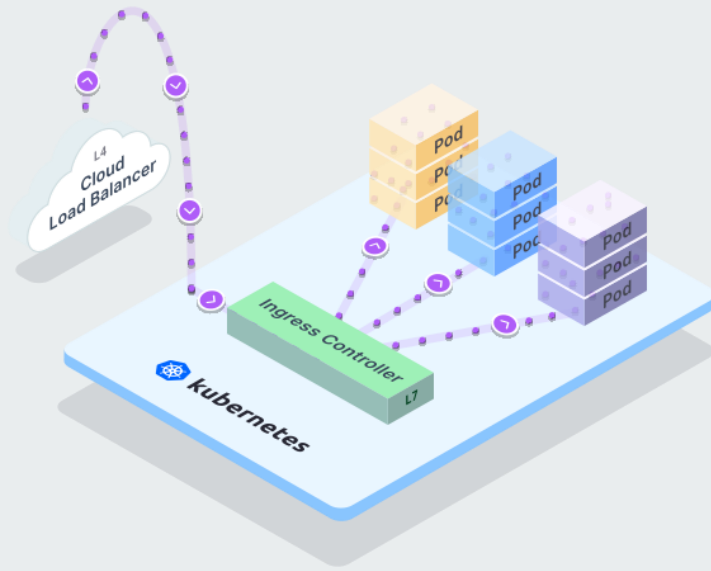


Kubernetes

Networking & Services

lamdaongc@gmail.com



Module Target

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Hiểu được các khái niệm cơ bản về khái niệm service trong Kubernetes
- Có khả năng phân biệt các loại service và tính ứng dụng của chúng
- Thực hành tạo các loại service nhằm cung cấp kết nối tới ứng dụng chạy trên môi trường Kubernetes
- Giới thiệu về khái niệm Ingress, Ingress Controller và khả năng ứng dụng

Outline

1 Kubernetes Networking Overview

- Đặt bài toán về kết nối mạng khi sử dụng Kubernetes
- Giới thiệu khái niệm Service và Endpoints trong Kubernetes

2 Kubernetes Service

- Các loại Service trong Kubernetes
- Phạm vi hoạt động của mỗi loại service

3 Ingress & Ingress Controller

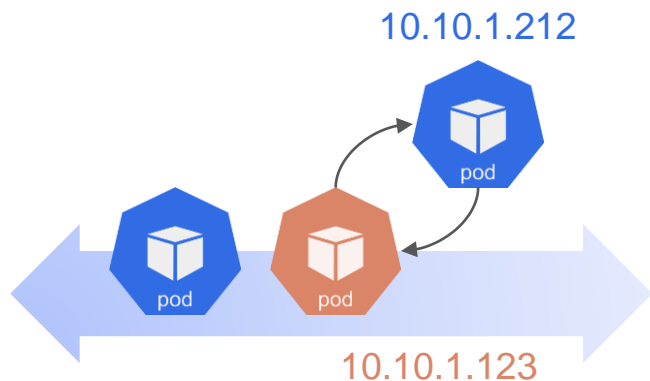
- Giới thiệu về khái niệm ingress và ingress controller
- Nguyên lý hoạt động của Ingress Controller và ứng dụng

Section 1:

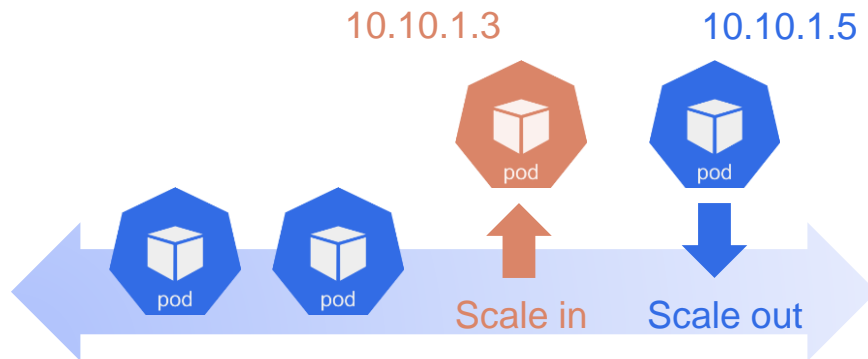
Kubernetes Networking Overview

Networking In Kubernetes

Địa chỉ IP của Pod có thể thay đổi do nhiều yếu tố:



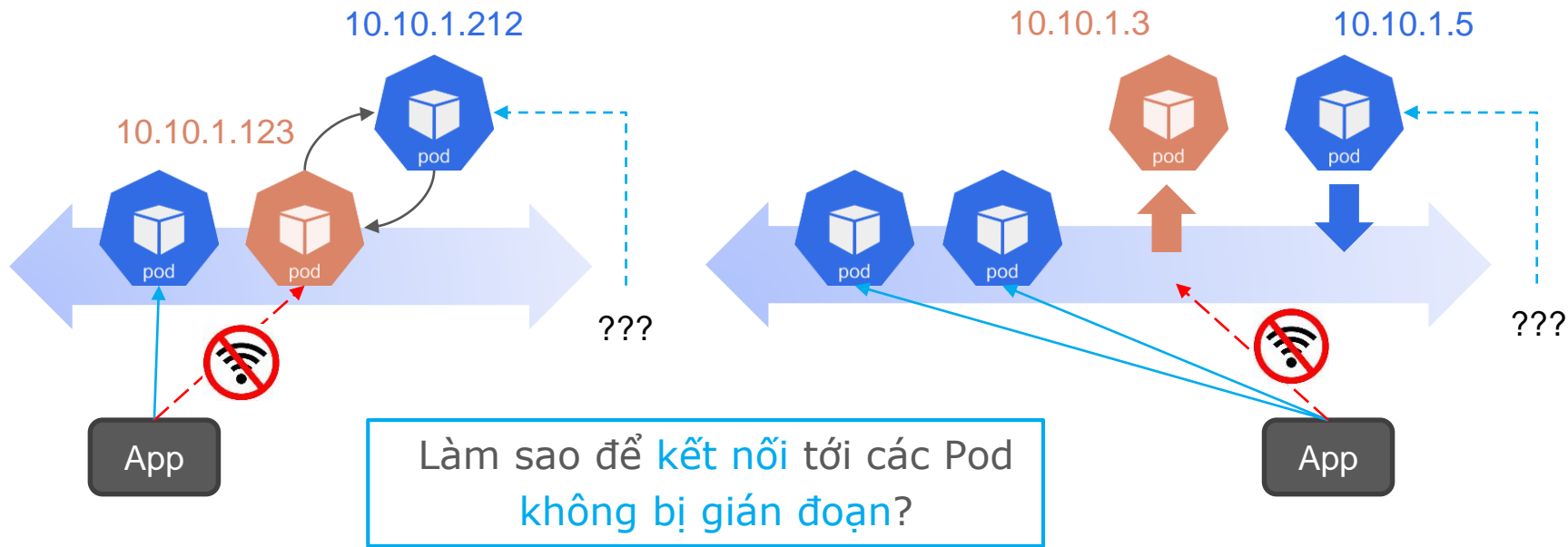
1 Quá trình **auto healing/upgrade** ứng dụng



2 Quá trình co giãn theo chiều ngang **scale in/scale out**

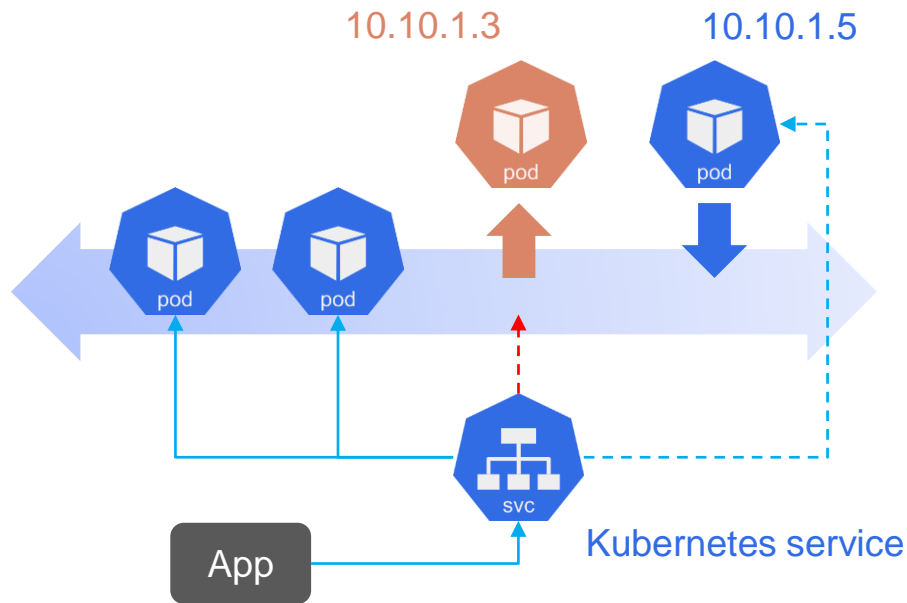
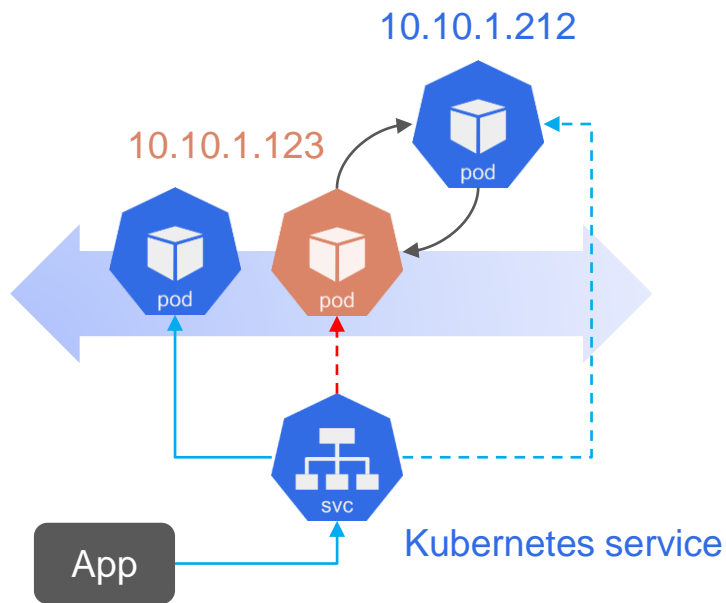
Networking In Kubernetes

Địa chỉ IP của Pod thay đổi khiến cho ứng dụng gặp **khó khăn khi kết nối** tới các Pod



Kubernetes Service

Giải pháp: Sử dụng **Kubernetes Service** nhằm cung cấp **1 endpoint duy nhất**, cho phép kết nối tới các Pod



Kubernetes Service & Endpoint



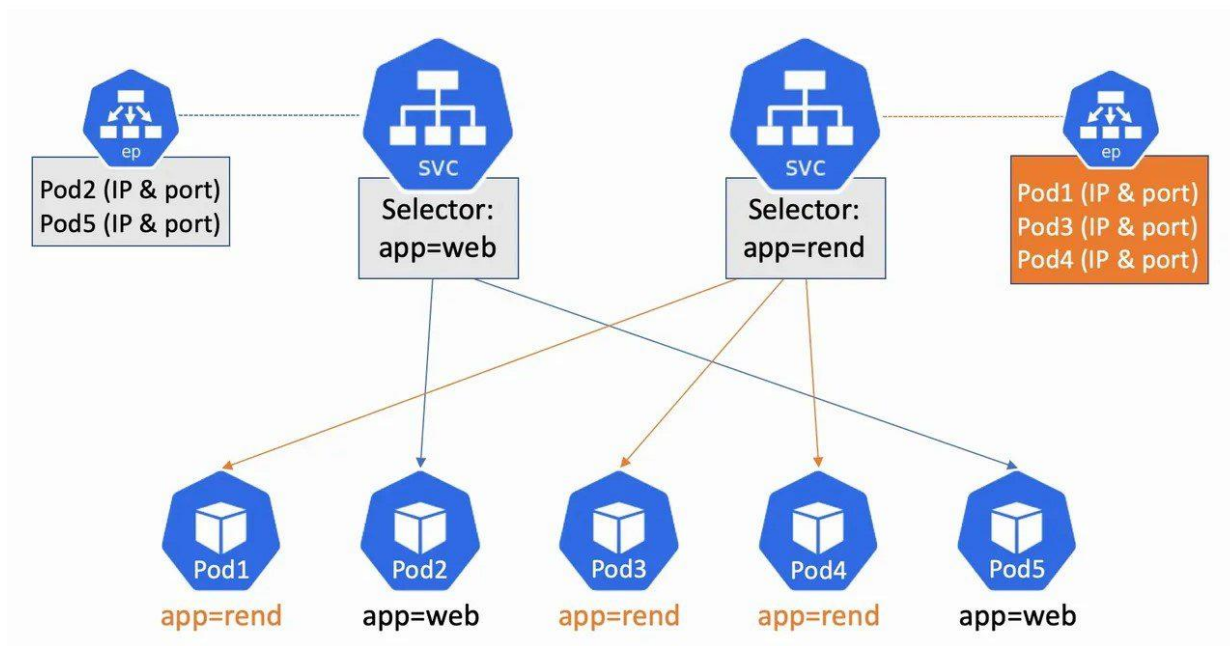
Service



Endpoint

- Service cung cấp một điểm kết nối duy nhất đại diện cho một danh sách các Pod
- Danh sách các Pod đứng sau Service được gọi là Endpoint
 - Danh sách các endpoint (pod ip address) được cập nhật liên tục
 - Service cung cấp tính năng healthcheck nhằm xác định trạng thái hoạt động của pod
- Một số loại Service được hỗ trợ bao gồm: ClusterIP, NodePort, LoadBalancer, ExternalName, Headless

Endpoint Selector



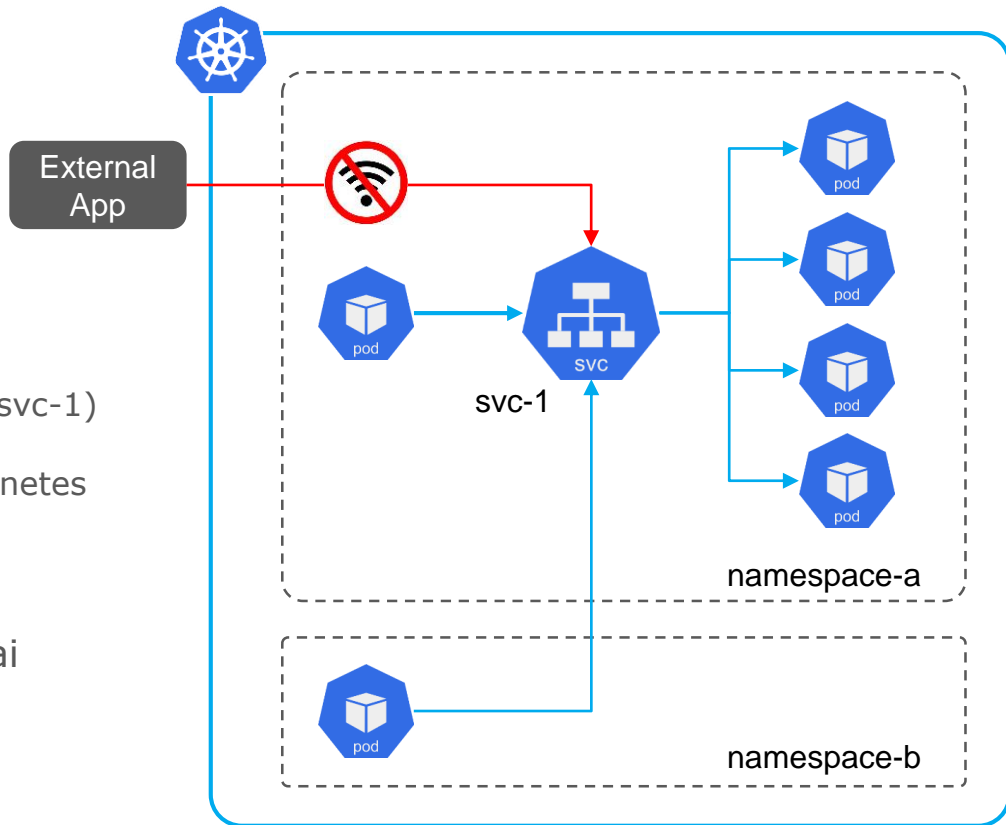
Section 2:

Kubernetes Services

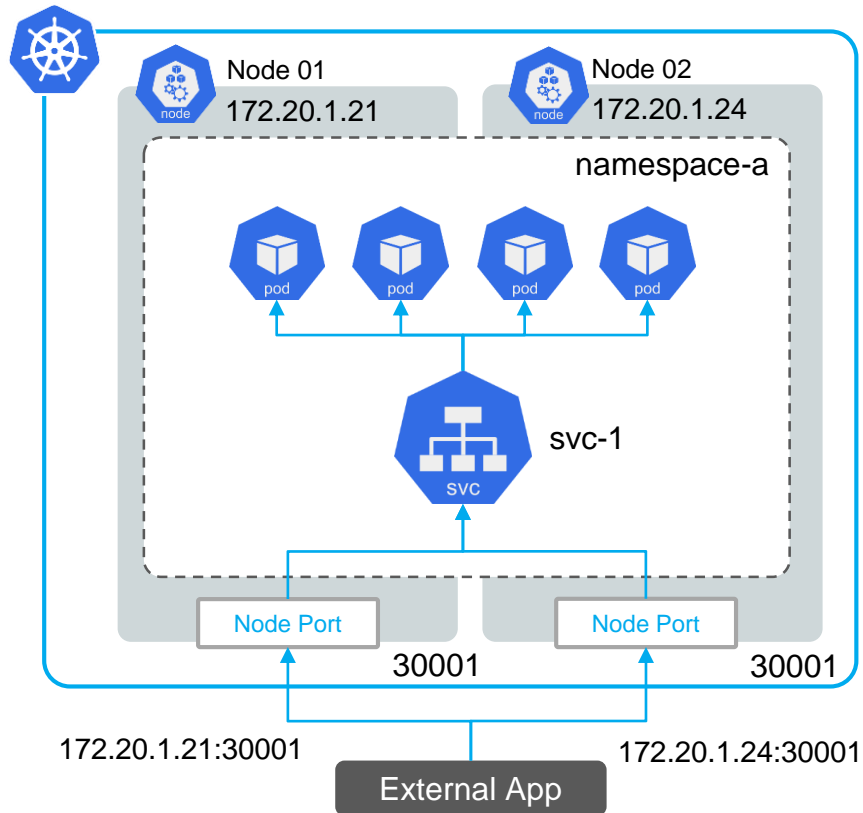
Service: ClusterIP

Loại service có giới hạn kết nối
nội bộ bên trong cluster

- Có khả năng tạo các kết nối nội bộ
 - Cùng namespace: gọi thẳng tên service (svc-1)
 - Khác namespace: gọi theo dns của Kubernetes (svc-1.namespace-b.svc.cluster.local)
- Không thể tạo các kết nối từ bên ngoài cluster tới service dạng clusterIP



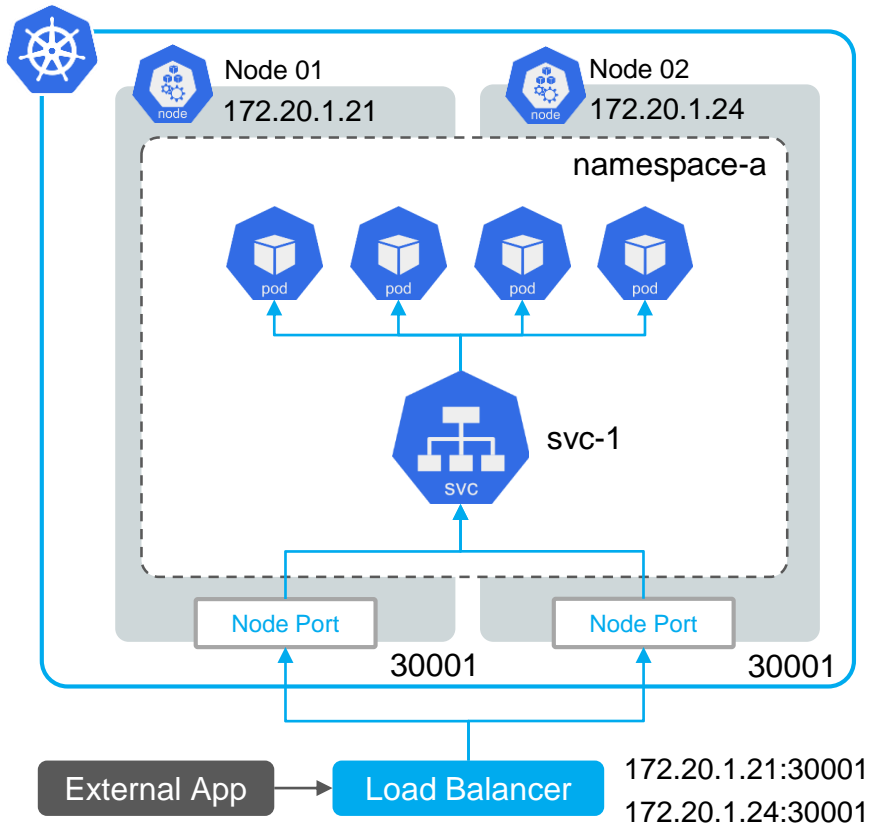
Service: NodePort



Cho phép tạo các kết nối tới service bằng cách mapping port trên node với service

- Có khả năng tạo các kết nối nội bộ tương tự như clusterIp
- Cho phép các kết nối từ bên ngoài cluster tới service thông qua địa chỉ: `<node-ip>:<node-port>`

Service: LoadBalancer



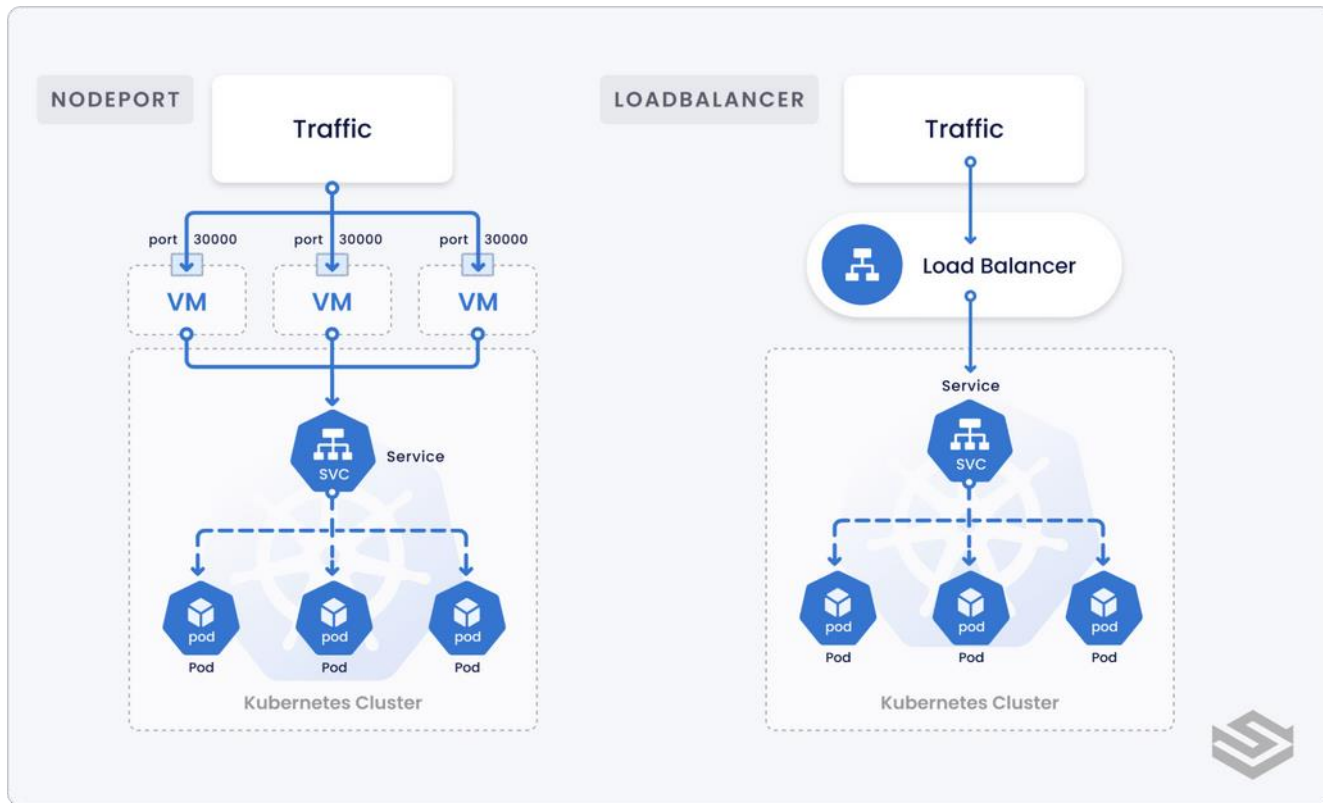
Đặt vấn đề:

làm sao để External App cập nhật được danh sách Node trong quá trình node scaling

loadBalancer Service

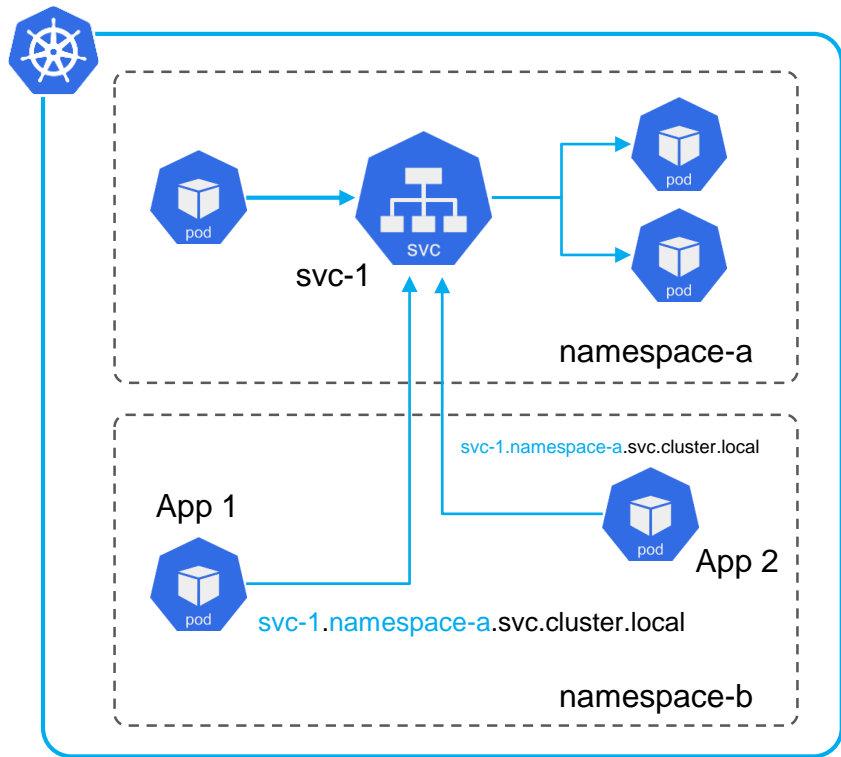
- Tạo ra 1 endpoint duy nhất giúp kết nối tới service từ bên ngoài cluster
- Sử dụng dịch vụ load balancer của cloud provider (AWS, GCP, Alibaba,...)
- Danh sách địa chỉ IP của các Node được tự động cập nhật

NodePort vs LoadBalancer

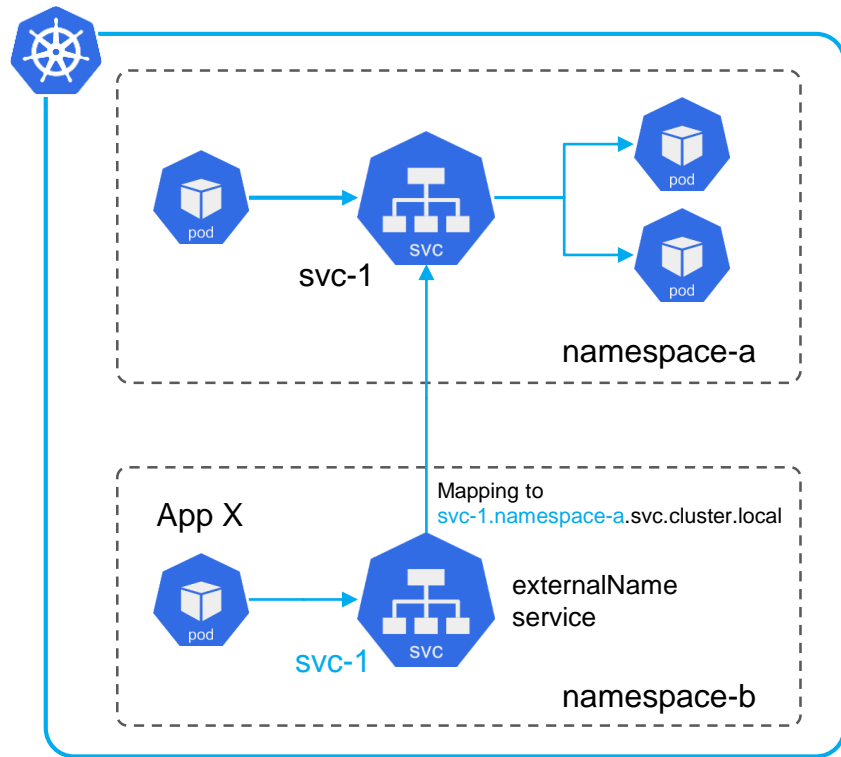


Service: ExternalName

From this...



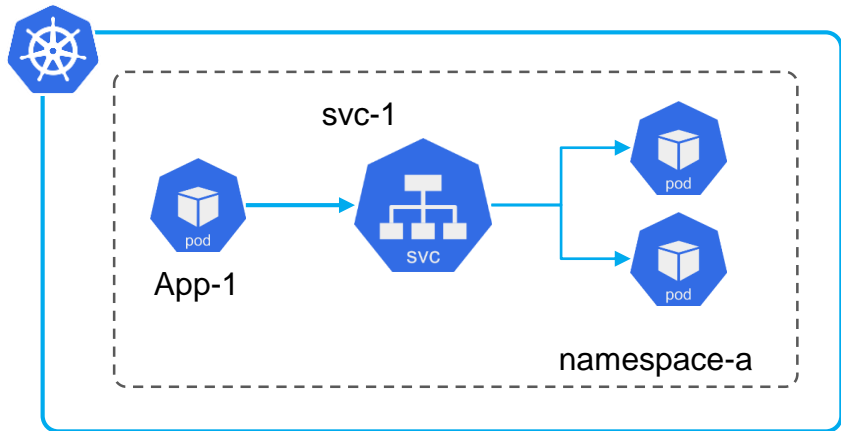
To this...



Service: Headless

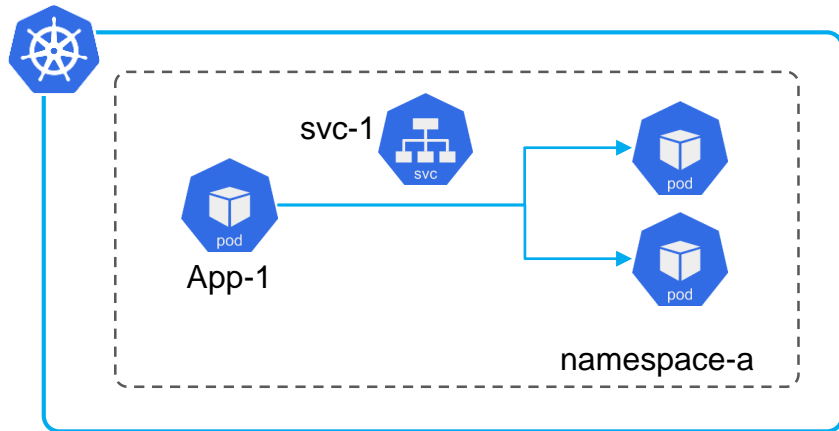
clusterIP service

- Kết nối tới các Pod thông qua service
- Các tác vụ như draining connection được service lo liệu



Headless service

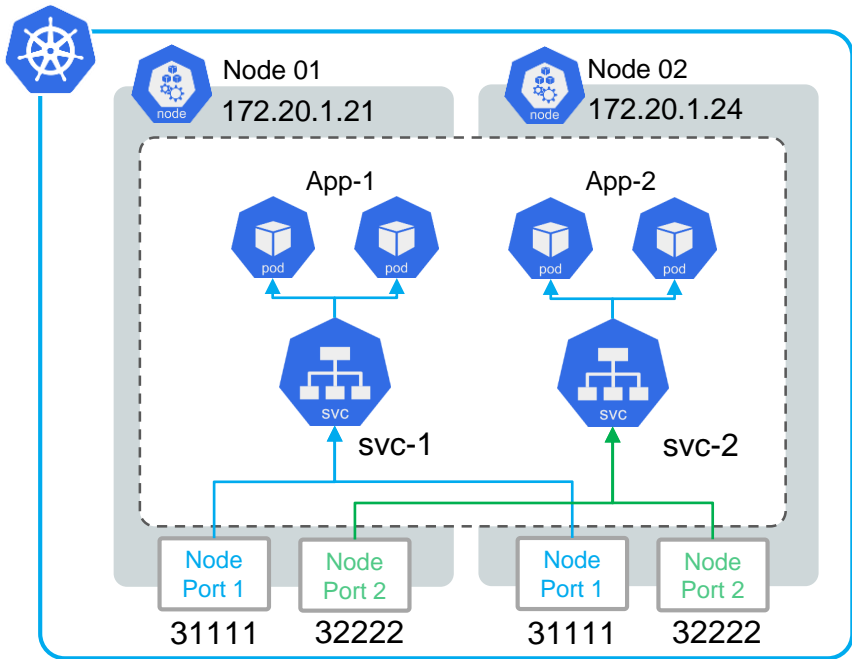
- Kết nối trực tiếp tới địa chỉ IP của Pod
- Ứng dụng App-1 phải tự xử lý các kết nối khi có update



Section 3:

Ingress Controller

Scenario 1:



Connect to App-1:
+ 172.20.1.21:31111
+ 172.20.1.24:31111

External App

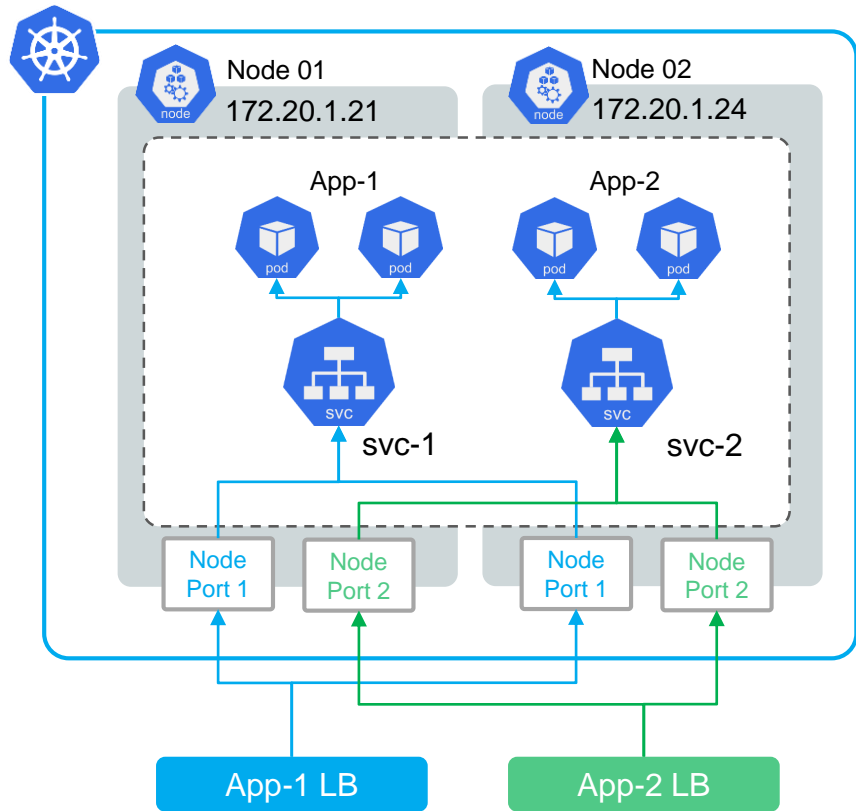
Connect to App-2:
+ 172.20.1.21:32222
+ 172.20.1.24:32222



Maintain multiple nodePort

- Việc quản lý trở nên khó khăn hơn khi triển khai quá nhiều service NodePort
- Ứng dụng phải cấu hình một loạt danh sách địa chỉ IP và port
- Vấn đề về khả năng tự động cập nhật danh sách các endpoint trong quá trình auto scaling/auto healing

Scenario 2:



Load Balancing cost

- Vấn đề về cập nhật danh sách endpoint có thể được giải quyết thông qua loadBalancer service
- Tuy nhiên phương án sử dụng nhiều loadBalancer service có thể khiến phát sinh chi phí hạ tầng

Ingress Controller

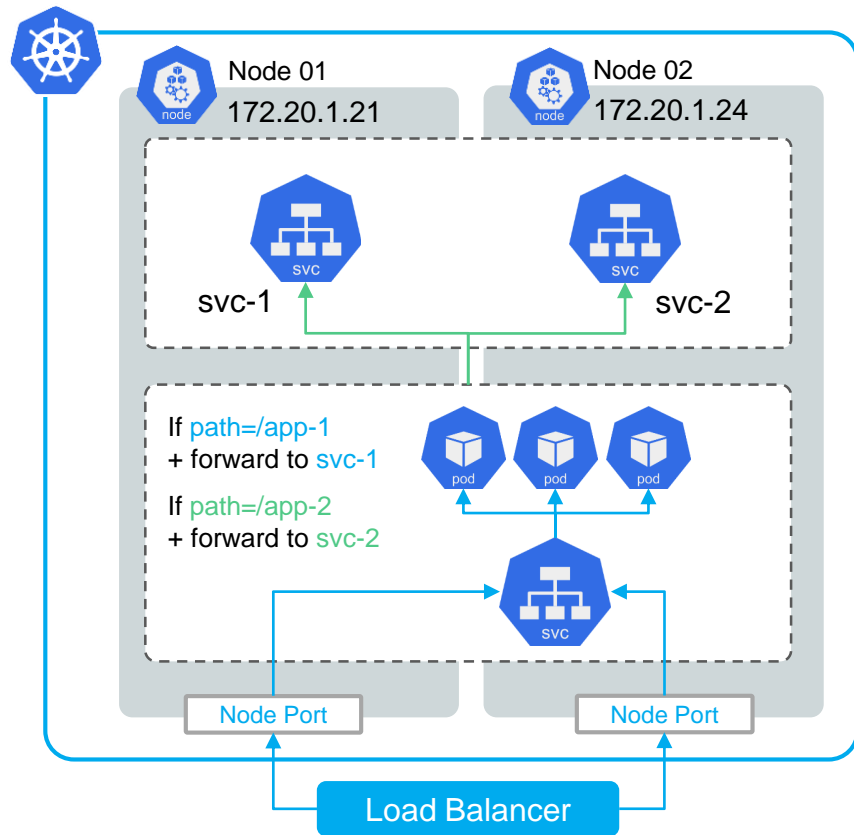
Ingress Controller

→ Thành phần đóng vai trò điều phối traffic tới các service khác trong cluster

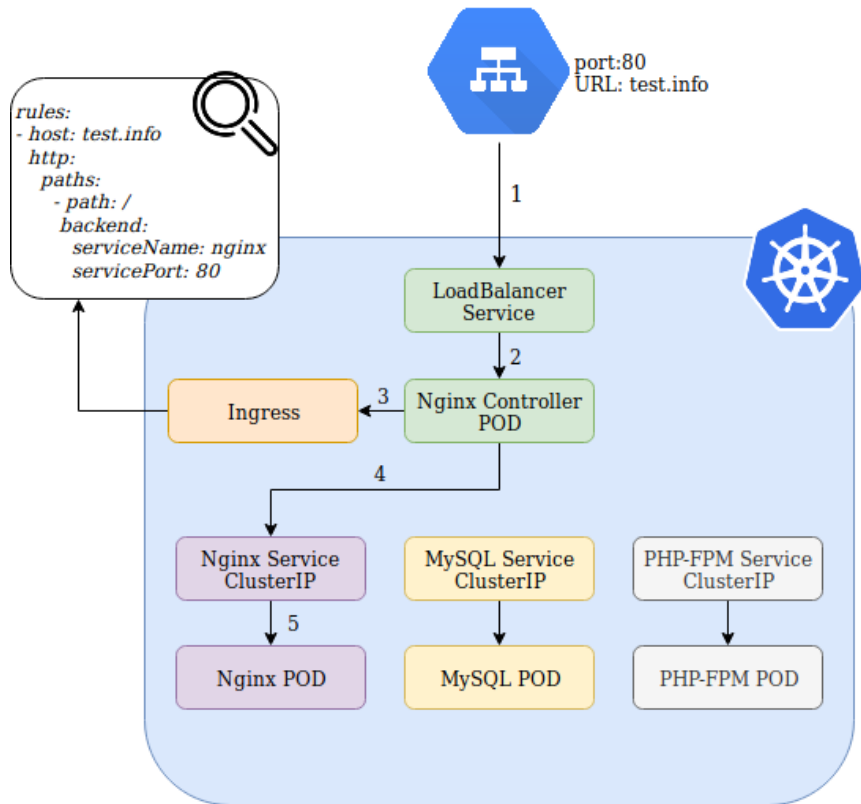
Một số Ingress controller thông dụng:
Nginx, HA-proxy Traefix,...

Ingress

→ Các quy tắc điều phối traffic được Ingress Controller sử dụng



Example: Nginx Ingress Controller



1. Người dùng truy cập test.info với path mặc định "/"
2. Truy cập thông qua LoadBalancer tới Nginx Ingress Controller Pod
3. Nginx Controller Pod thực hiện kiểm tra các cấu hình "Ingress"
4. Traffic được chuyển tiếp tới Nginx Service ClusterIP
5. Cuối cùng, traffic tới được Nginx Pod

Section 4:

Summary

Module Target

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Hiểu được các khái niệm cơ bản về khái niệm service trong Kubernetes
- Có khả năng phân biệt các loại service và tính ứng dụng của chúng
- Thực hành tạo các loại service nhằm cung cấp kết nối tới ứng dụng chạy trên môi trường Kubernetes
- Giới thiệu về khái niệm Ingress, Ingress Controller và khả năng ứng dụng

Thank you