

Triển khai API Gateway với Kong trên Kubernetes

Nguyễn Hữu Kim - @DevSuccess101

- Nguyễn Hữu Kim - Technical Leader / Researcher
- @ Sun* Vietnam, R&D Unit - 8 years
- GitHub / Facebook / Twitter: [@kimyvgy](#)
- Rock, karaoke, liên quân, chạy bộ...
- Dev Success 101 - <https://devsuccess101.com>
 - Youtube, TikTok, Viblo
- Nhờ mọi người follow ủng hộ nhé! ❤️ ❤️ ❤️



Agenda

1. Tìm hiểu về Kong Gateway

- API Gateway là gì?
- Kong Gateway hoạt động như nào?
- Demo Kong với Docker Compose

2. Kong Gateway trên Kubernetes

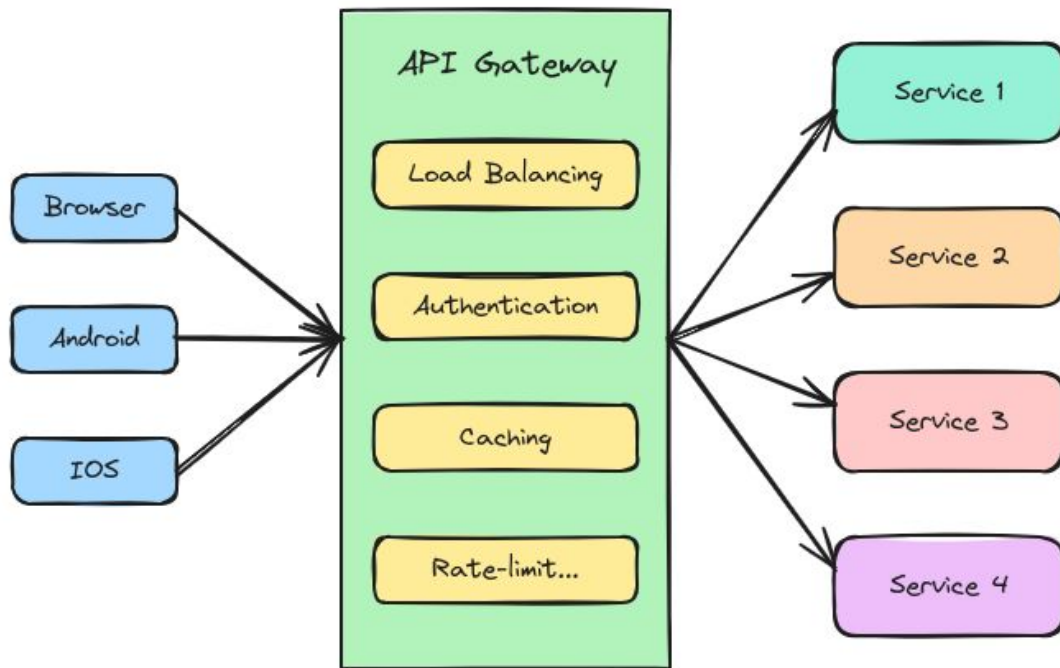
- Service, Ingress
- Ingress Controller
- Kong Ingress hoạt động như nào?
- Demo Kong Ingress với K8s

3. Question & Answer

1. Kong - API Gateway

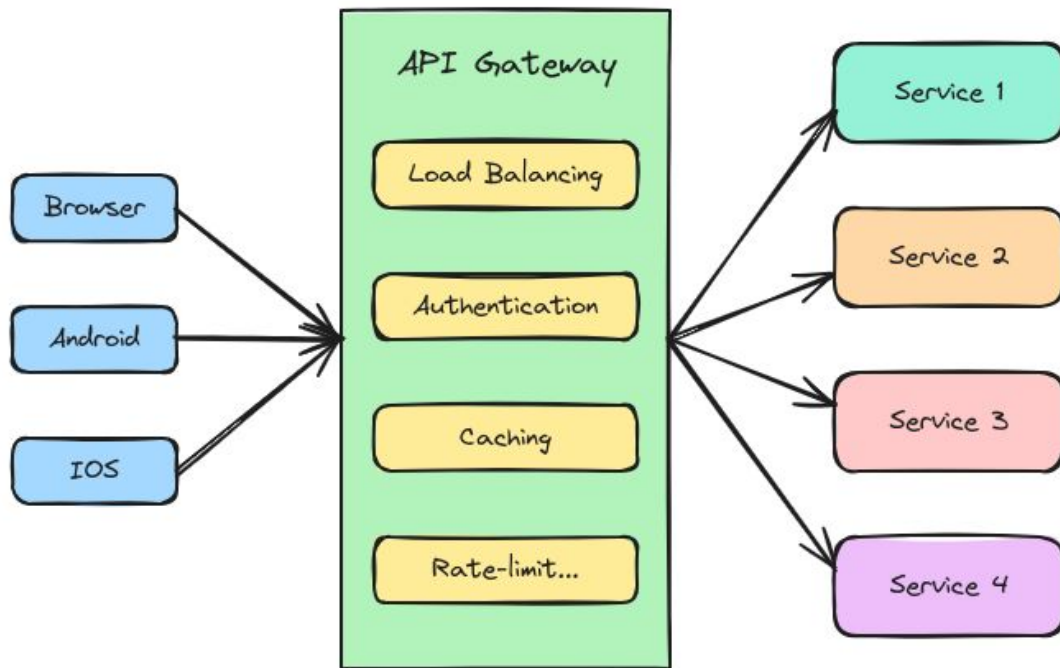
API Gateway là gì?

- API Gateway là cổng truy cập (entrypoint) của một hệ thống phân tán.
- Đứng giữa client và các backend service.
- Reverse proxy, load balancer.



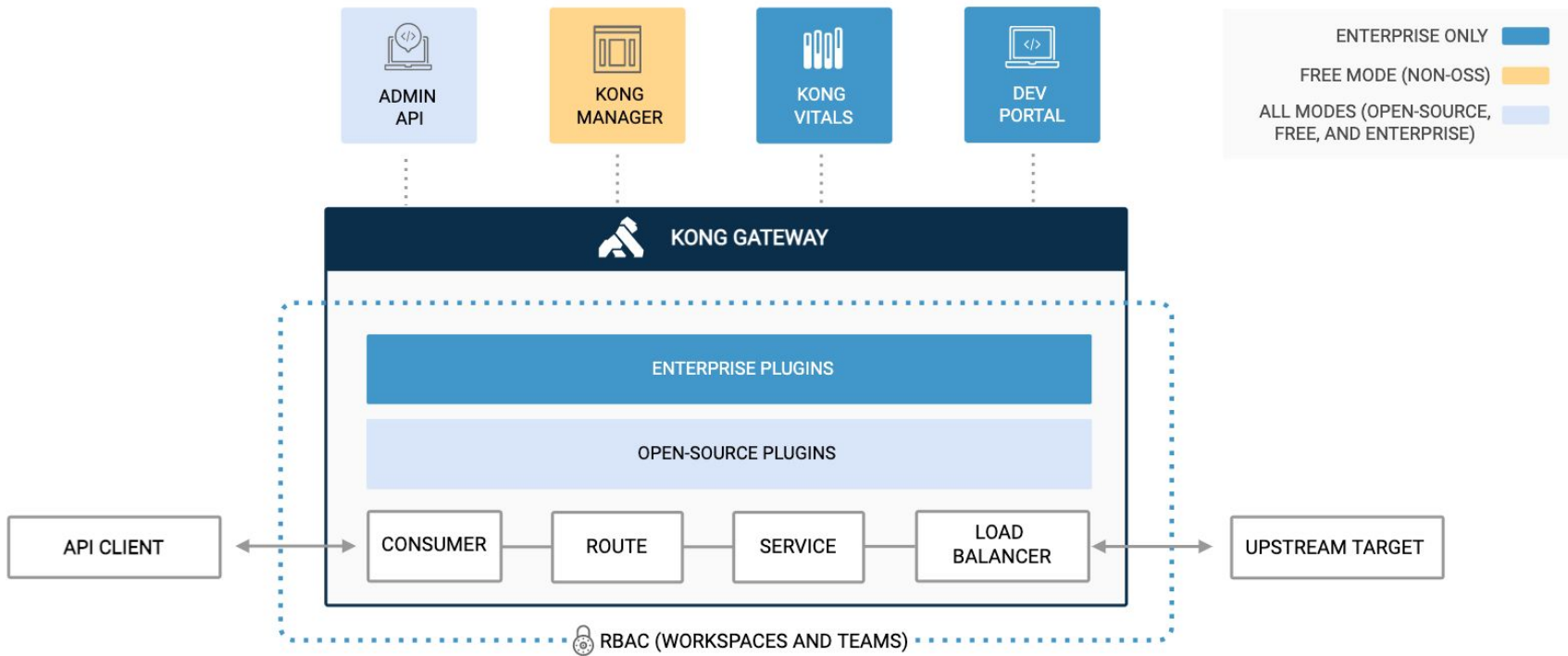
API Gateway là gì?

- Cung cấp nhiều chức năng hoàn chỉnh, cần thiết để quản lý APIs:
 - Authentication
 - Authorization
 - Rate-limit, Routing
 - Monitoring, Alert...
- API Gateway vs Microservices



- Giảm độ phức tạp:
 - Một endpoint có khả năng scale
 - Giảm code cho những tính năng chung: authentication, authorization, rate-limit, caching...
 - Dễ dàng giám sát, kiểm soát truy cập, theo dõi
- Tối ưu performance:
 - Load Balancing tới các backend service
 - Hỗ trợ gRPC, JSON API
 - Filter các request không hợp lệ, caching giúp giảm tải cho backend service
- An toàn hơn:
 - Tích hợp nhiều cơ chế authentication
 - Filter IP và các cơ chế secure

Kong Gateway hoạt động như nào?

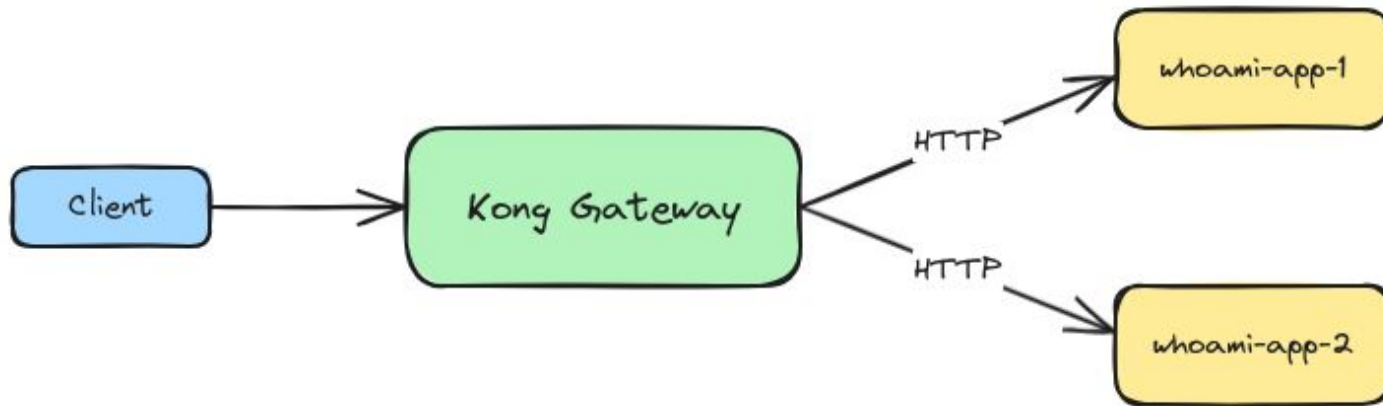


Kong Gateway hoạt động như nào?

- **Upstream target:** Service đích cần thực hiện cân bằng tải
- **Load balancer:** Module điểm nhiệm cân bằng tải
- **Service:** Đại diện cho các microservices, các API mà Kong quản lý
- **Route:** Định nghĩa các quy tắc để match client requests đến service tương ứng
- **Consumer:** Đại diện đối tượng sử dụng dịch vụ (human, machine), sử dụng cho authentication
- **API Client:** Là bất kỳ ứng dụng nào muốn truy cập tới API được quản lý bởi Kong

Hands-on: Kong Gateway trên Docker

Demo: Kong Gateway với Docker Compose



- Kong Gateway v3.5 + PostgreSQL: <http://localhost:8000>, <http://localhost:8001>
- Kong Manager: <http://localhost:8002>
- Thực hiện routing & load balancing cho 2 container **whoami-app-1** và **whoami-app-2**.
- Truy cập vào qua browser để kiểm tra: <http://whoami.lc:8000/api/v1>

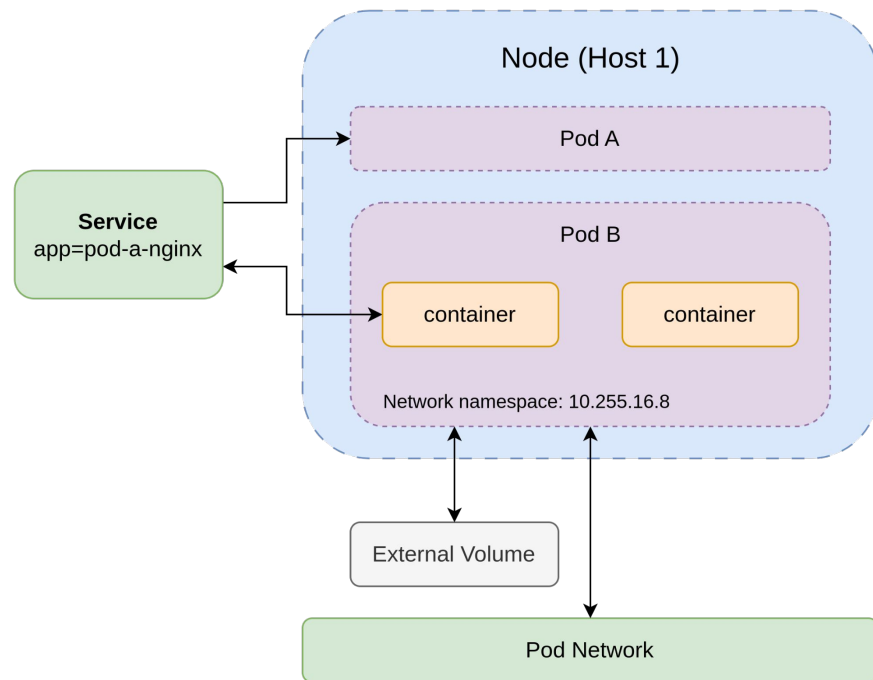
2. Kong Gateway trên Kubernetes

Hiểu về Service, Ingress trong Kubernetes

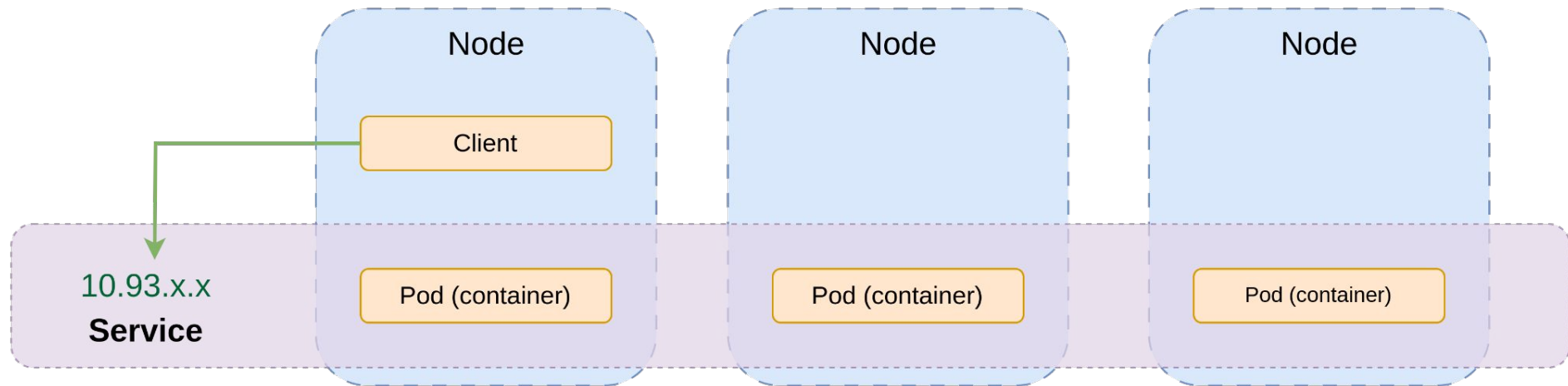
- K8s Terminology:
 - Service
 - Ingress
 - Ingress Controller

Service trong Kubernetes

- Pod là thành phần cơ bản nhất trong Kubernetes cluster.
- Các Pod được ảo hóa và cô lập hoàn toàn với môi trường xung quanh, kể cả container ở hai Pod khác nhau cũng **không thể giao tiếp với nhau**.
- Service là thành phần cầu nối, đóng vai trò expose container để các ứng khác trong cluster có thể tiếp được với nhau.

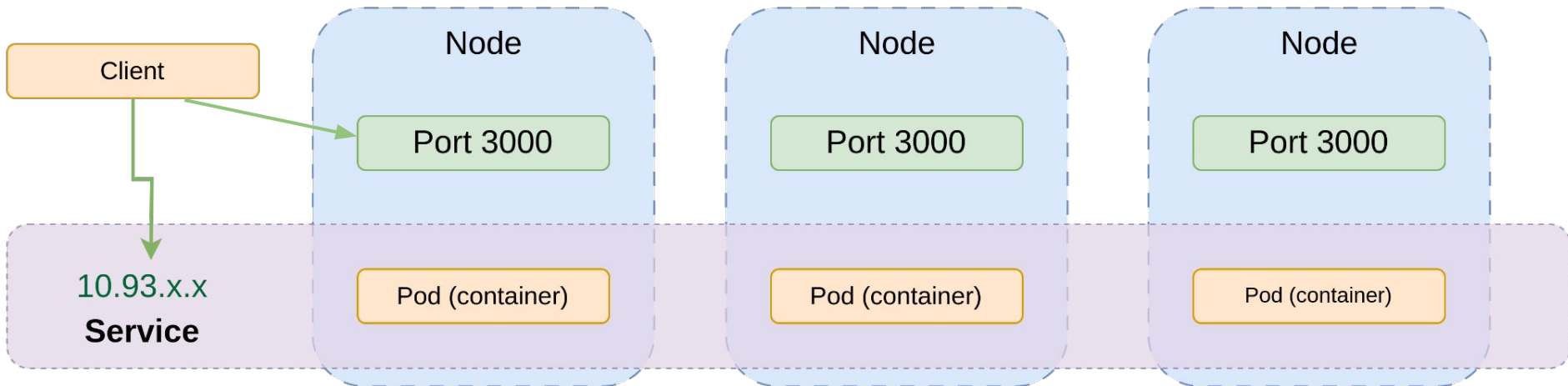


- ClusterIP Service thực hiện **expose application** trong mạng nội bộ của cluster.



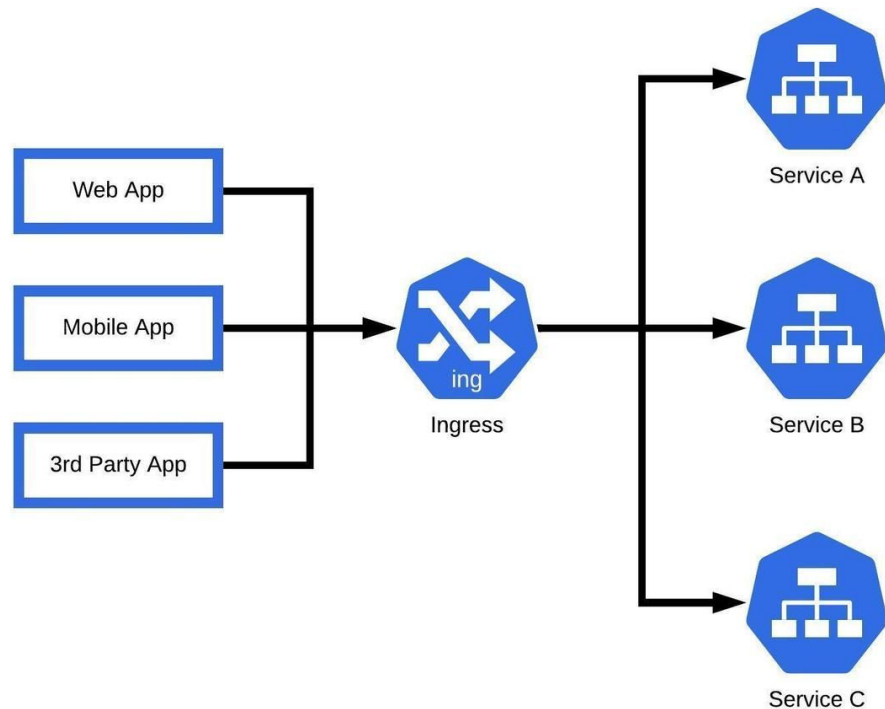
NodePort Service

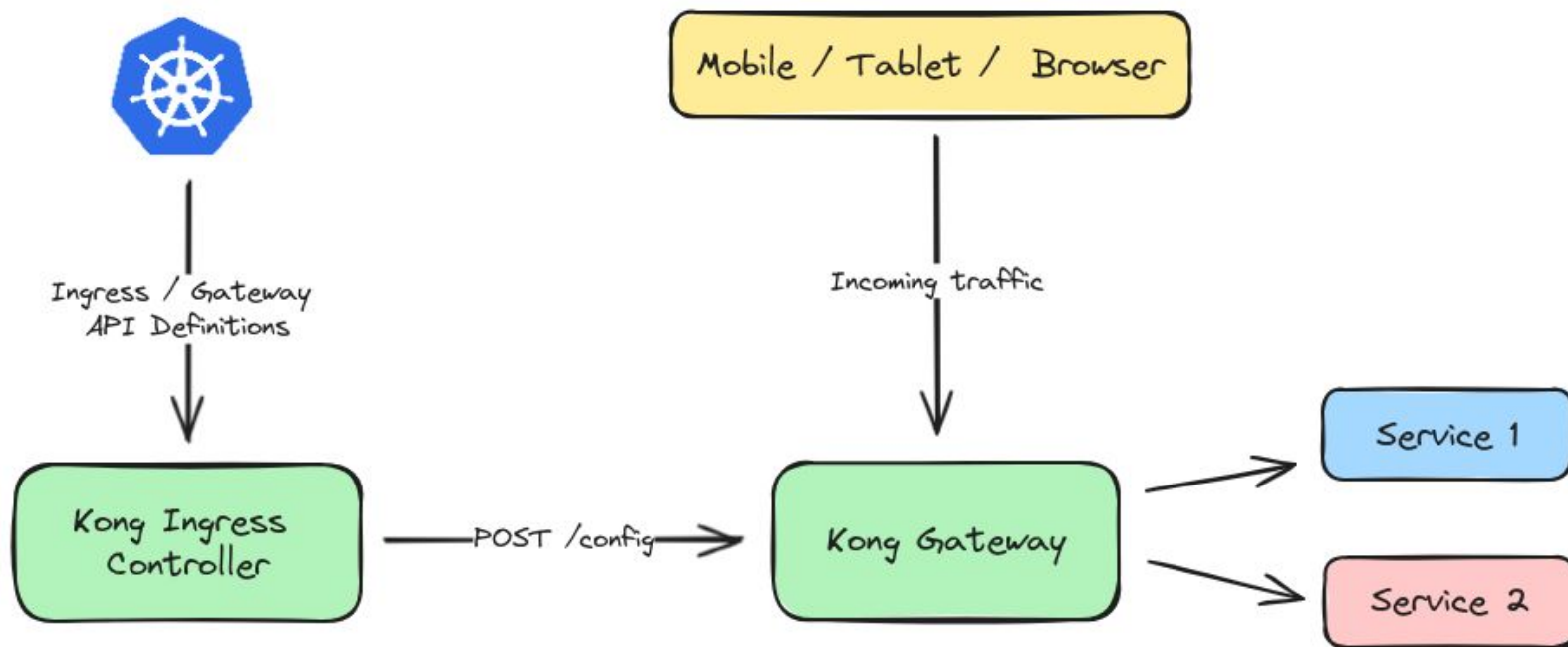
- NodePort Service thực hiện **expose application ra internet** bằng cách listen một external port trên mỗi node.



Ingress là gì?

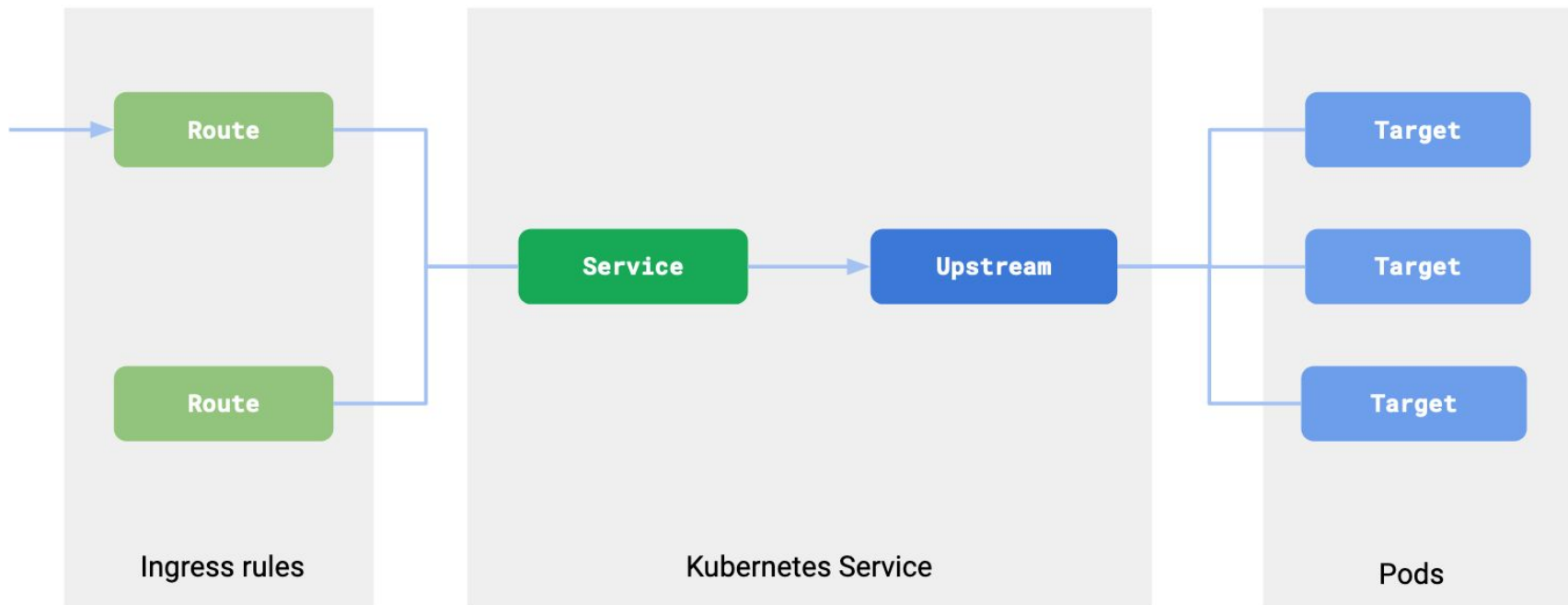
- Ingress là một Kubernetes object giúp quản lý truy cập đến các Services **từ bên ngoài cluster**.
- Chung mục đích với NodePort và Load Balancer.
- Cung cấp thêm nhiều tính năng mở rộng:
 - HTTPS
 - Basic authentication...
- Luôn hoạt động K8s Service và cần phải có một **Ingress Controller**.





Cách thức hoạt động của Kong Ingress Controller

=> **Kong Gateway** sẽ nhận traffic,
không phải Kong Ingress Controller



Services và Routes của Kong trên Kubernetes

Hands-on: Kong Gateway trên Kubernetes

- Tài liệu liên quan:
 - [Helm.sh](#)
 - [Install KIC - v3.x](#)
 - [Viblo](#)
- Sử dụng helm chart:
 - [kong/ingress](#)
 - <https://charts.konghq.com>
 - DBLess, Custom Resources

```
apiVersion: configuration.konghq.com/v1
kind: KongConsumer
metadata:
  name: alice
  annotations:
    kubernetes.io/ingress.class: "kong"
username: alice
credentials:
- alice-key

---

kind: Secret
apiVersion: v1
metadata:
  name: alice-key
stringData:
  key: bylkogdatomoryakom
  kongCredType: key-auth
```

```
~/d/k/kong-ingress *master> curl http://localhost:8000/api/v1/auth/me -v
* Trying 127.0.0.1:8000...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 8000 (#0)
> GET /api/v1/auth/me HTTP/1.1
> Host: localhost:8000
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404 Not Found
< Date: Thu, 14 Dec 2023 19:43:11 GMT
< Content-Type: application/json; charset=utf-8
< Connection: keep-alive
< Content-Length: 103
< X-Kong-Response-Latency: 0
< Server: kong/3.5.0
< X-Kong-Request-Id: c403e759cfe1ca535112da117cec5675
<
{
  "message": "no Route matched with those values",
  "request_id": "c403e759cfe1ca535112da117cec5675"
}
* Connection #0 to host localhost left intact
```

```
~/d/k/kong-ingress *master> curl -H 'Apikey: bylkogdatomoryakom' \
-H 'Host: whoami.k8s.lc' \
http://localhost:8000/api/v1/auth/me
Hostname: whoami-6554f4d754-rtgfn
IP: 127.0.0.1
IP: 10.244.1.60
RemoteAddr: 10.244.1.67:53454
GET /api/v1/auth/me HTTP/1.1
Host: whoami.k8s.lc
User-Agent: curl/7.68.0
Accept: */*
Apikey: bylkogdatomoryakom
Connection: keep-alive
X-Consumer-Id: b32379d9-3d28-5683-a79c-72139be6e34f
X-Consumer-Username: alice
X-Credential-Identifier: bffd0b28-366c-41ab-adfc-3168d8ab623e
X-Forwarded-For: 127.0.0.1
X-Forwarded-Host: whoami.k8s.lc
X-Forwarded-Path: /api/v1/auth/me
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Kong-Request-Id: faac286e0447c84e636f1e5eaab5eb26
X-Real-IP: 127.0.0.1
```

Q&A

Thank you for watching!

Please raise your hand up for asking question.

Subscribe! @DevSuccess101

