# Interface trong các hệ thống phân tán

### Câu hỏi 1: Interface trong hệ thống phân tán là gì? Tại sao cần phải có Interface khi triển khai các dịch vụ?

* Interface (giao diện) là tập hợp các quy ước chuẩn mà cả bên cung cấp dịch vụ lẫn bên sử dụng dịch vụ đều phải tuân thủ. Interface quy định cách thức trao đổi thông điệp, cú pháp lệnh, ngữ nghĩa của phản hồi, và phiên bản của giao thức
* Nhờ có Interface, các hệ thống khác nhau có thể tương tác với nhau một cách thống nhất, tránh tình trạng “mạnh ai nấy làm”, dễ dàng mở rộng, thêm các dịch vụ hiện có mà không cần thay đổi hệ thống, ngoài ra Interface còn giúp dễ dàng quản lý và xử lý lỗi giữa các thành phần.

### Câu hỏi 2: Hãy giải thích ý nghĩa của mã trạng thái 201 CREATED, 204 NO\_CONTENT và 404 NOT\_FOUND trong giao thức KVSS

* 201 CREATED: mã miêu tả kết quả phản hồi từ server khi client put thành công cặp Key-Value.
* 404 NOT\_FOUND: mã miêu tả kết quả phản hồi từ server khi client get giá trị của key không tồn tại.

### Thực hành Mini Key-Value Store (KVSS)

#### Tổng quan Server:

* Xây dựng được Server TCP, multithreading (mỗi thread xử lý 1 client riêng biệt)
* Lưu trữ KV trong dictionary
* Ghi log ra console, file theo đúng định dạng
* Hoạt động đúng theo logic, đúng theo mô tả interface với 9 test case

#### Tổng quan Client:

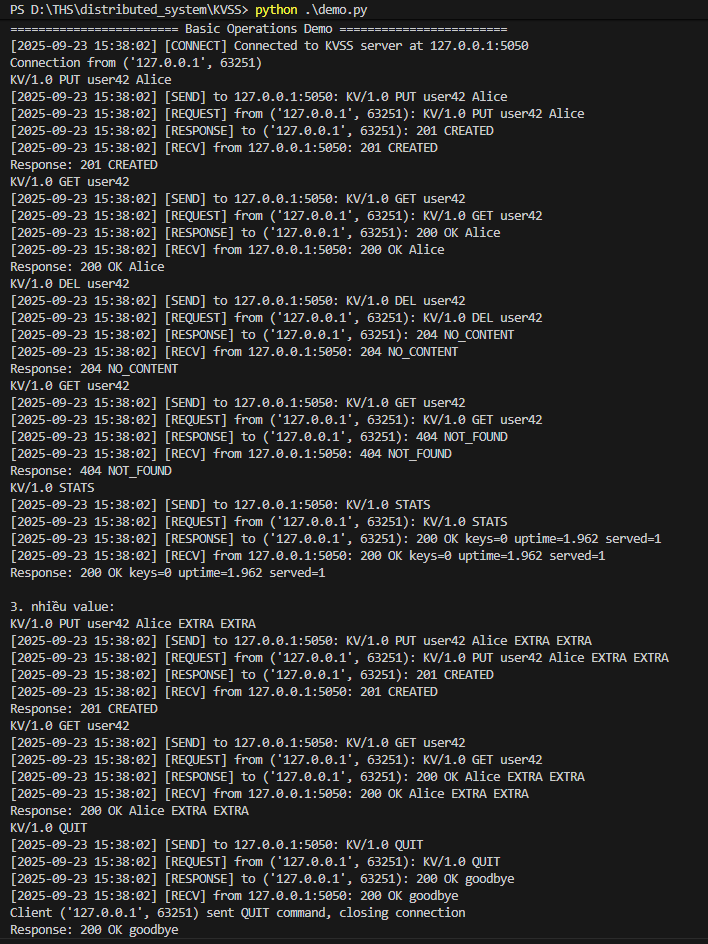
* Client dòng lệnh, đọc stdin; gửi request theo chuẩn; in nguyên văn status + data.
* Ghi log ra console, file theo đúng định dạng

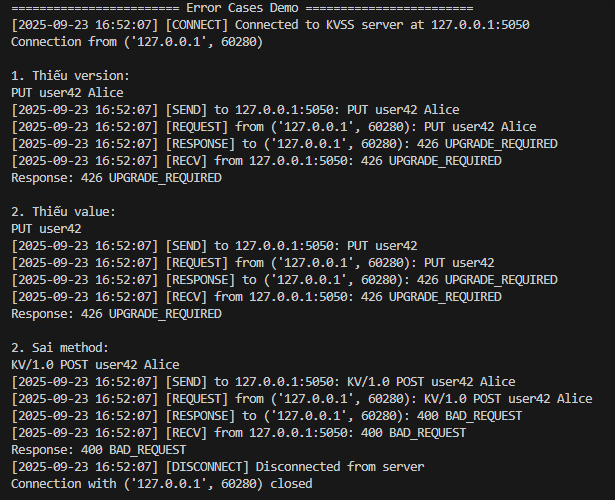
#### Source code:

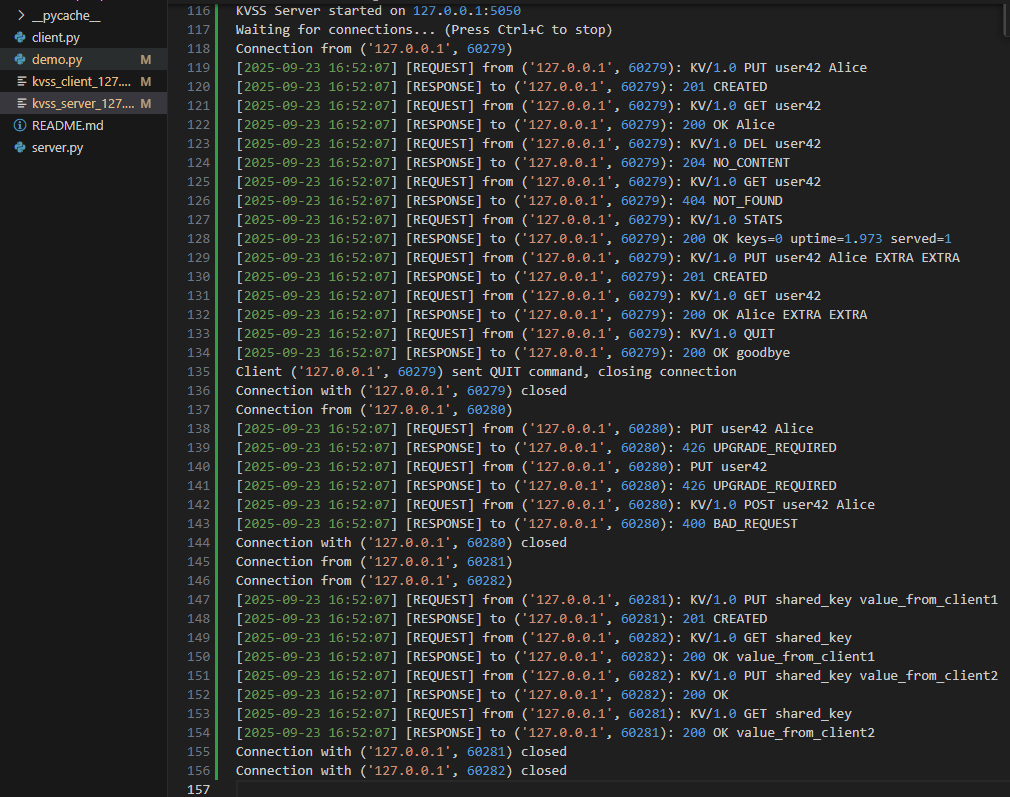
* https://github.com/hieuxyzt/distributed\_system

#### Demo test case:

Test các ví dụ trong file bài tập của thầy và thêm trường hợp “KV/1.0 PUT user42 Alice EXTRA EXTRA”





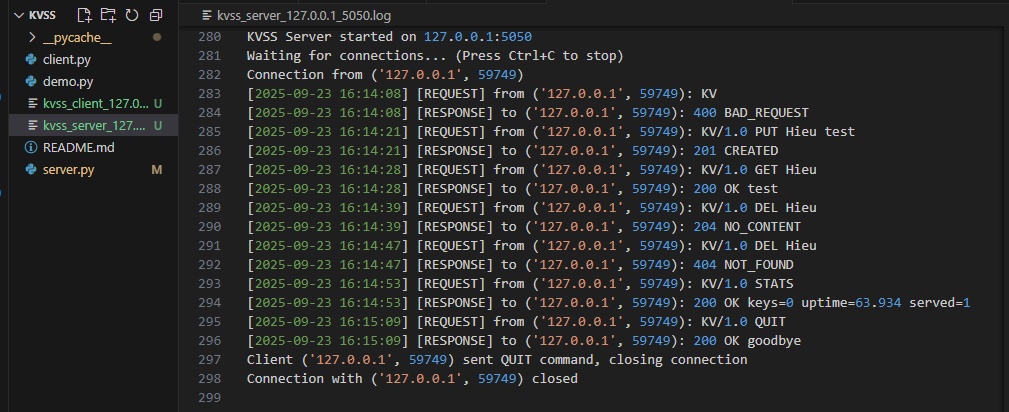


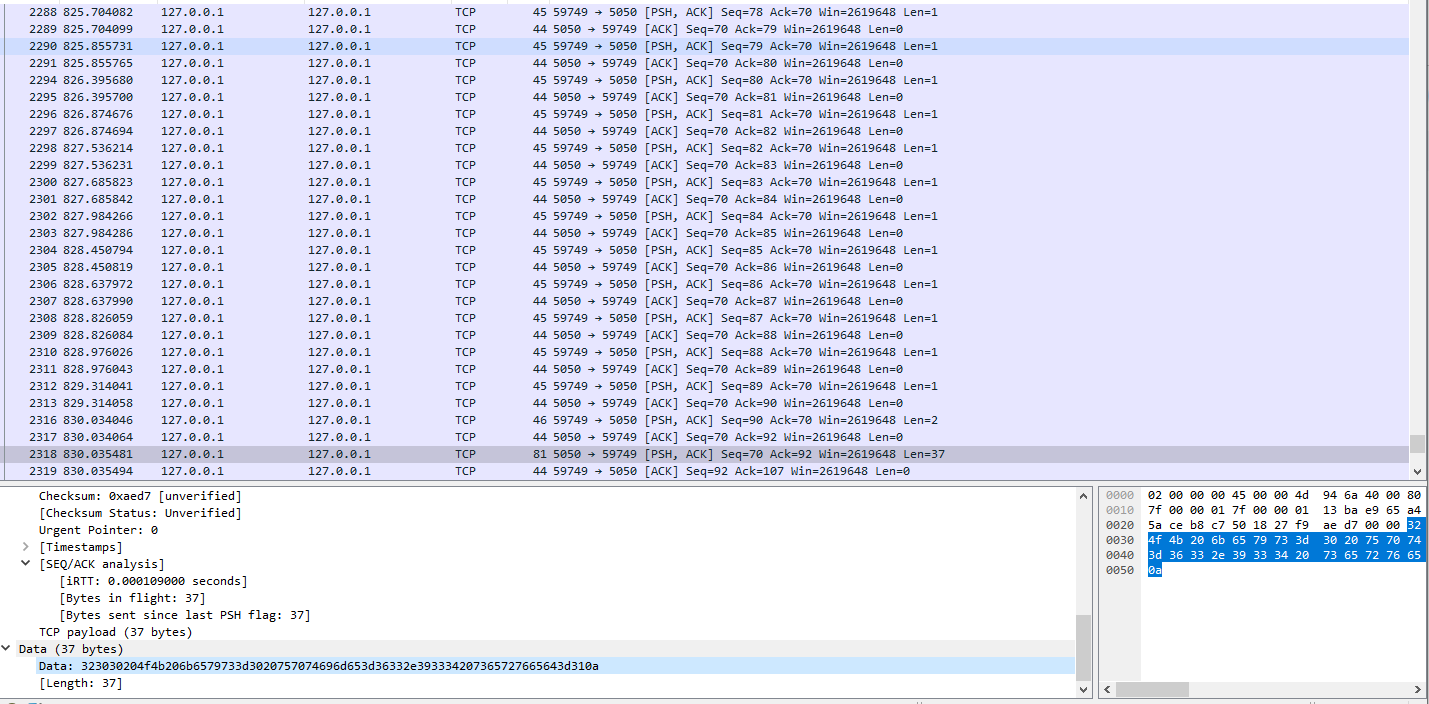
#### Demo sử dụng telnet và wireshark:

Cài đặt wireshark, capture loopback traffic, filter: tcp.port == 5050

telnet localhost 5050





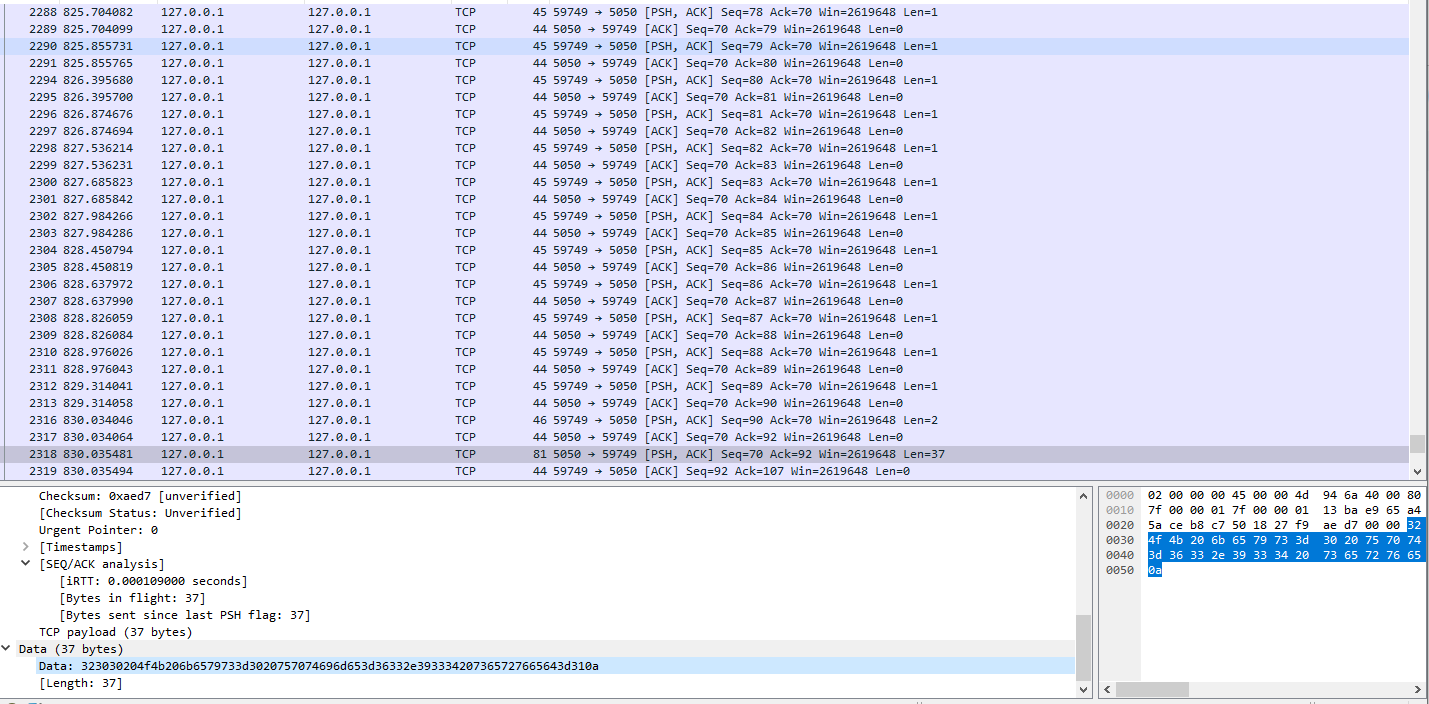


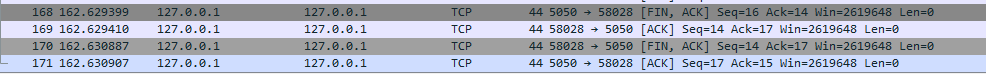
### Câu hỏi 3: Trong bài lab KVSS, nếu client không tuân thủ quy ước Interface (ví dụ: thiếu version KV/1.0), server sẽ phản hồi thế nào? Tại sao phải quy định rõ ràng tình huống này?

* Server sẽ trả lại mã lỗi riêng theo từng trường hợp. Ví dụ thiếu version KV/1.0, server trả về “426 UPGRADE\_REQUIRED” tương đương thiếu version hoặc phải nâng cấp version.
* Phải quy định rõ ràng tình huống thiếu version này vì tùy theo từng version, **format** của request/response hoặc **logic** xử lý có thể cho kết quả khác nhau.

### Câu hỏi 4: Quan sát một phiên làm việc qua Wireshark: hãy mô tả cách mà gói tin TCP được chia để truyền thông điệp theo “line-based protocol”.



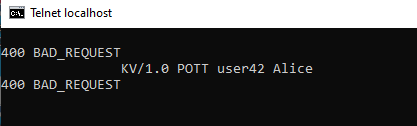




Từ kết quả tracing gói tin từ wireshark, ta thấy:

* Khi thực hiện telnet, client với server mở kết nối TCP thông qua quy trình bắt tay ba bước (SYN, SYN/ACK, ACK)
* Khi gõ câu lệnh, các ký tự được gửi lần lượt đến server (14 ký tự, \n chiếm 2 ký tự) tương đương với câu lệnh “KV/1.0 STATS\n”.
* Server sau khi nhận được “\n” sẽ thực thi và trả lại response “0000 32 30 30 20 4f 4b 20 6b 65 79 73 3d 30 20 75 70 0010 74 69 6d 65 3d 36 33 2e 39 33 34 20 73 65 72 76 0020 65 64 3d 31 0a” hex tương đương với “200 OK keys=0 uptime=63.934 served=1” dạng utf-8.
* Sau khi client gửi câu lệnh ngắt kết nối “KV/1.0 QUIT”, quy trình ngắt kết nối 2 chiều diễn ra (FIN, ACK)

### Câu hỏi 5: Giả sử có một client viết sai giao thức (gửi KV/1.0 POTT user42 Alice). Server sẽ xử lý như thế nào? Kết quả này thể hiện đặc điểm gì của Interface?

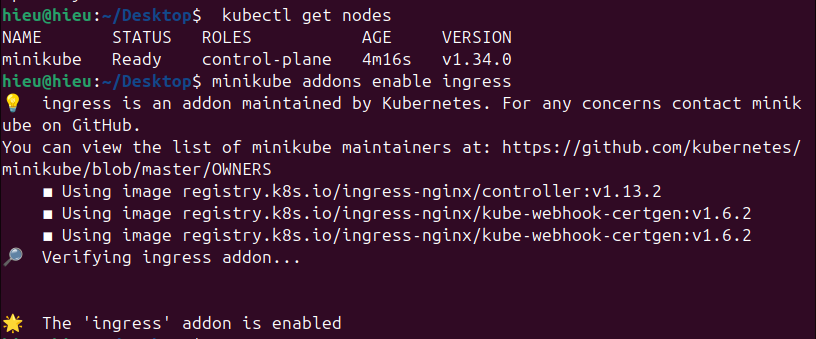


* Server sẽ check method POTT không tồn tại và trả về kết quả 400 BAD\_REQUEST
* Kết quả này thể hiện đặc điểm Validation (khả năng xác thực và kiểm tra tính hợp lệ của dữ liệu client gửi lên server)

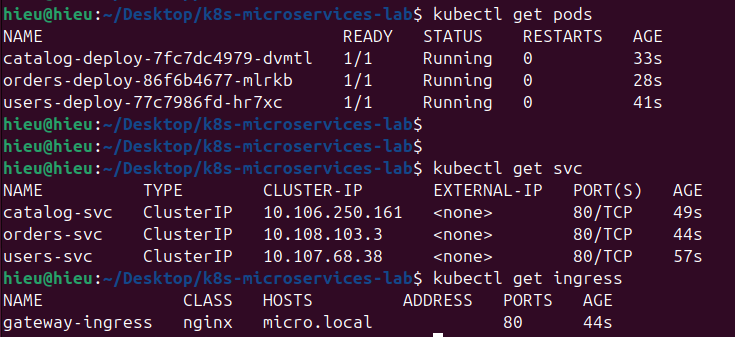
# Kiến trúc Microservices

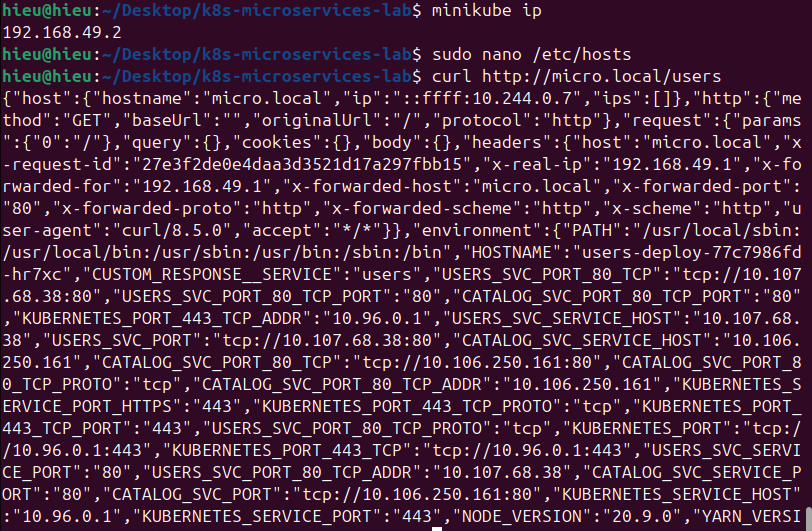
### Cài đặt môi trường

Hoàn tất cài đặt docker, minikube theo tài liệu



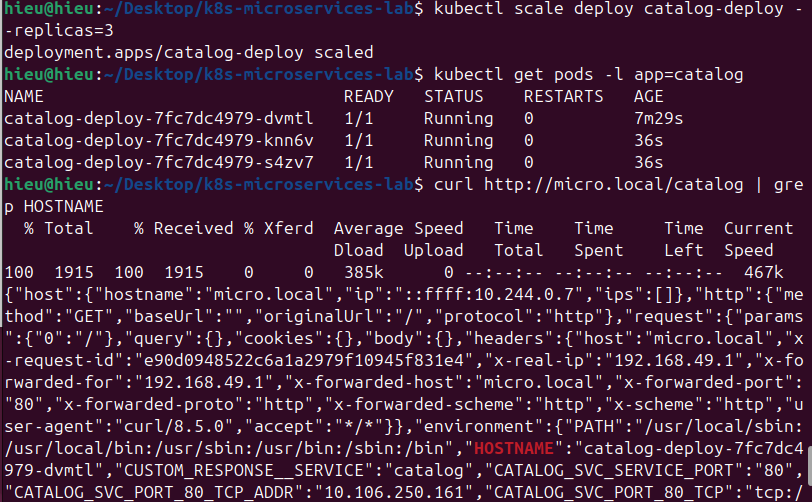
### Triển khai hệ thống





### Scale dịch vụ

Sau khi scale dịch vụ, khi gọi /catalog đã thấy hostname thay đổi: “catalog-deploy-7fc7dc4979-dvmtl”, “catalog-deploy-7fc7dc4979-s4zv7”, “catalog-deploy-7fc7dc4979-knn6v” đúng theo kịch bản cân bằng tải

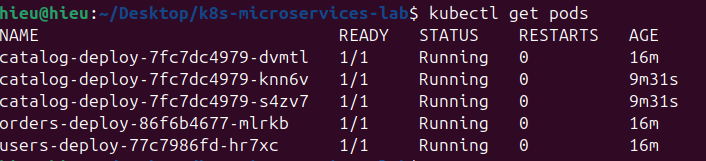






### Câu hỏi 6: Sau khi chạy kubectl apply -f users-deploy.yaml, dùng lệnh nào để kiểm tra Pod của service users đã chạy thành công? Hãy chụp màn hình kết quả

Sử dụng lệnh “kubectl get pods” để kiểm tra Pod service users đã chạy thành công



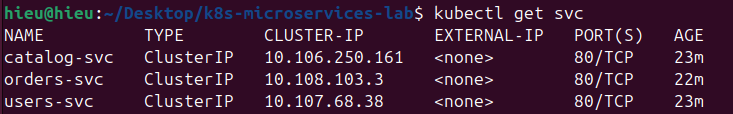
### Câu hỏi 7: Trong file users-deploy.yaml, hãy chỉ ra:

### - Deployment quản lý bao nhiêu replica ban đầu?

### - Service thuộc loại nào (ClusterIP, NodePort, LoadBalancer)?

Deployment quản lý 1 replica ban đầu dựa vào spec.replicas=1

Service thuộc loại ClusterId (mặc định) khi không config spec.type



### Câu hỏi 8: Sau khi cài Ingress, em cần thêm dòng nào vào file /etc/hosts để truy cập bằng tên miền micro.local?

