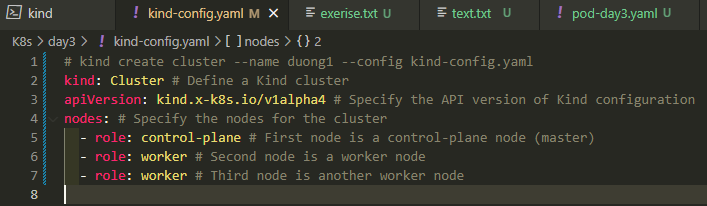
Day 3 exerise

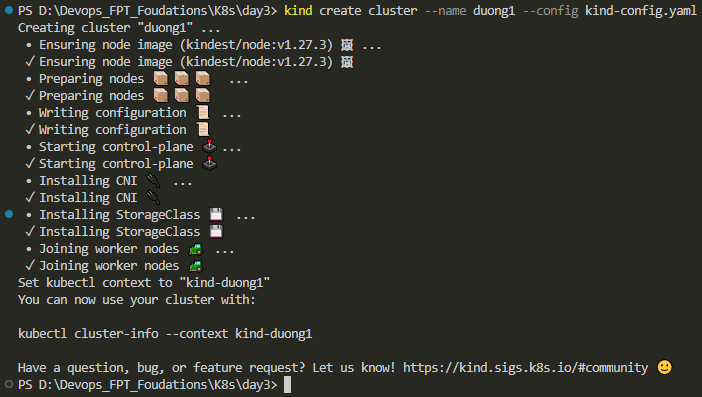
Bài 1: How many services in the cluster? Create 1 pod with nginx:alpine image and 1 pod

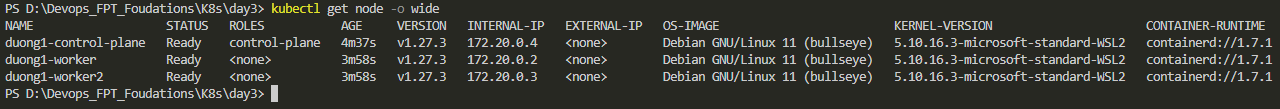
with httpd image and assign same labels to them

(mọi người thêm options --show-labels để nó hiện ra label của các pod xem có trùng ko nhé)

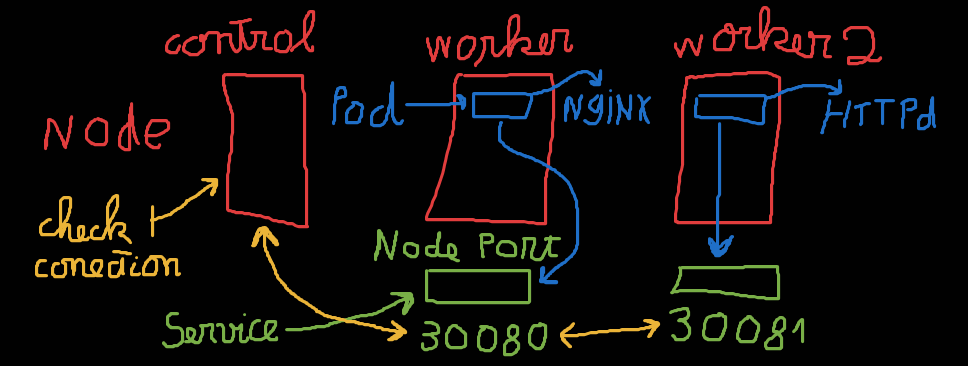
Trước hết chúng ta tạo Kind cluster với tên là duong1: (kind create clusters –name duong1) với file config kind-config.yaml (ta tạo 3 node trong đó 1 node là control plane và 2 node worker)



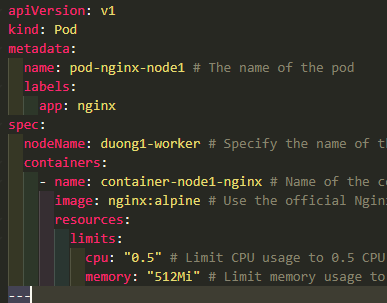
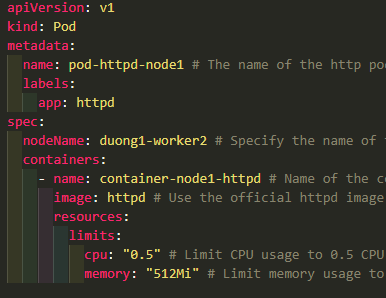




Với hình ảnh trên chúng ta đã tạo được 1 cluster và có 3 node

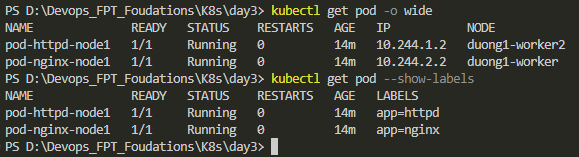
Ý tưởng ban đầu: 

Giờ chúng ta tạo 2 pod: pod 1 nginx trên worker1 và pod 2 httpd trên worker2 ta sẽ sau đó mở node port và check connection, hiện giờ ta viết file pod-day3.yaml để config 2 pod theo ý tưởng trên

Ta viết manifest cho 2 pod mỗi pod đều có conatiner chạy riêng nginx image và httpd image rồi cũng gán lables và giới hạn cho chúng tài nguyên cho 2 container chúng chạy

Ta nhập lệnh và check kết quả (ta cũng thêm opstion labels để xác nhận có trùng lables không)

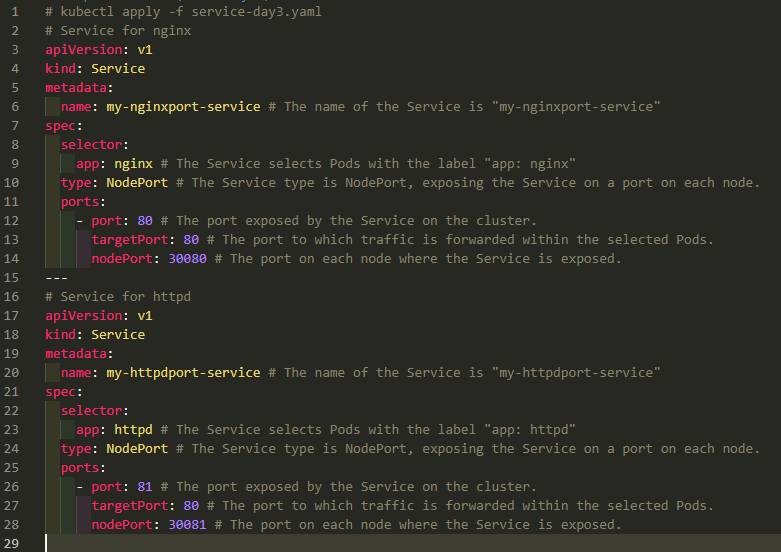


Trả lời câu hỏi How many services in the cluster?

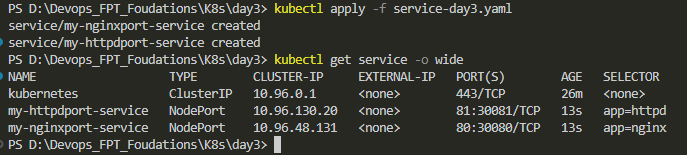
* There are four types of services that Kubernetes supports: ClusterIP, NodePort, LoadBalancer, and Ingress

Bài 2: Tạo 1 service nodeport có label khớp với 2 container vừa tạo, kubectl port-fortward

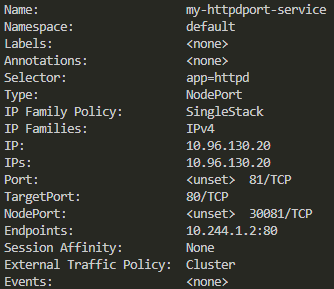
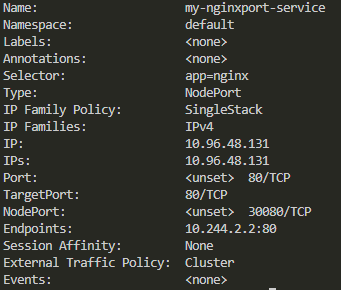
Đầu tiên chúng ta tạo file service-day3.yaml config để mở node port của worker 1 và worrker 2



Ta apply và chạy



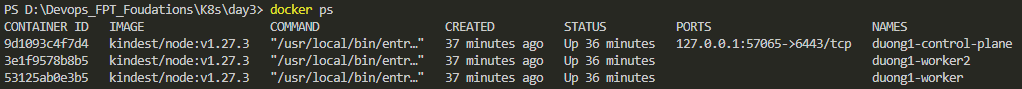
Ta check chi tiết end points của service httpd đã gán vào ip của httpd pod và ip container httpd 80

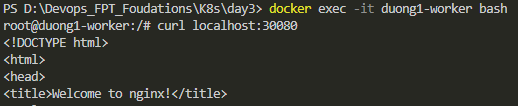
Ta tương tự check nginx pod và nginx container 80

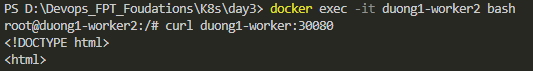


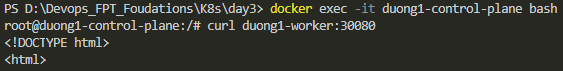
Ta check node của kind chạy trên docker container



Ta test connection của nginx pod chạy trên worker 1 với chính nó và các container khác có connect

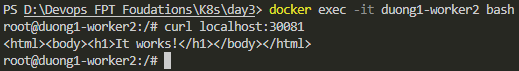
chính nó curl, nginx được hiện ra

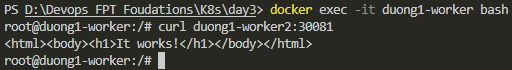
từ worker2 curl vào cổng nginx

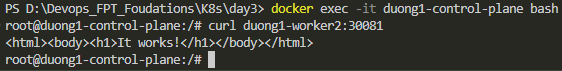
từ control-plane

Như vậy chúng ta thấy tất cả các node đều có thể connect được vào ip của nginx container được mở trên pod-nginx trong worker-node1 và chuyển hướng trafic bằng service node port tới cổng 80 của nginx

Ta test connection của httpd pod chạy trên worker 2 với chính nó và các container khác có connect

chính nó curl, httpd được hiện ra

từ worker 1 curl vào cổng httpd

 từ control-plane

Như vậy chúng ta thấy tất cả các node đều có thể connect được vào ip của httpd container được mở trên pod-httpd trong worker-node2 và chuyển hướng trafic bằng service node port tới cổng 80 của httpd