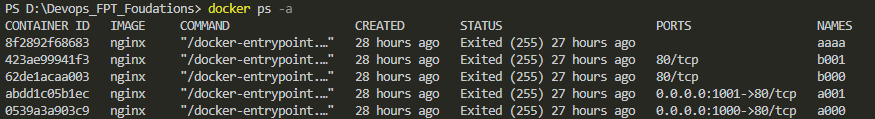
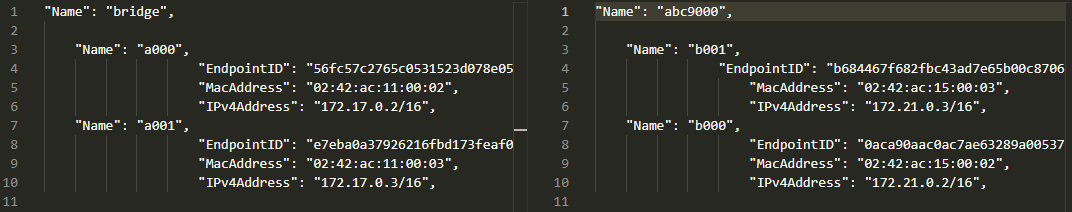
K8s – Day 2 exerise

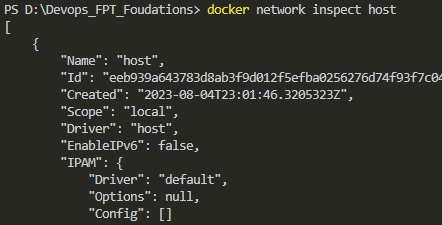
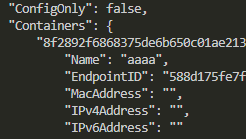
|  |
| --- |
| Task1 : docker tạo 3 container: 2 conter không config network -> mặc định bridge, 1 container config network khác (network mọi người tự tạo). Ping các container với nhau  Task2 : k8s tạo cluster 3 node, 4 pod. Ping các node và các pod với nhau |

Thứ nhất chúng ta tạo ra 5 nginx container với:

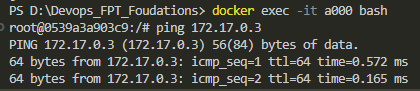
* 2 container với tên a000 và a001 không config network và sẽ được mặc định network bridge
* 2 container với tên b000 và b001 được cho vào network abc9000 với type là user bridge
* 1 contaner với tên aaaa được cho vào network host

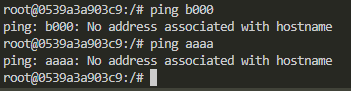




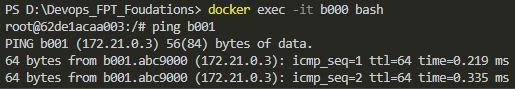
 

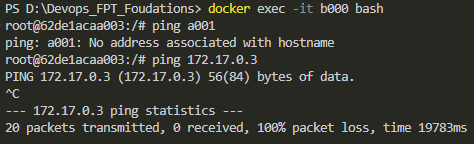
Chúng ta sẽ thử ping nhóm network default bridge với chính trong nhóm và ngoài nhóm

với ip 172.17.0.3 là ip của a001

không thể ping network abc9000 và network host

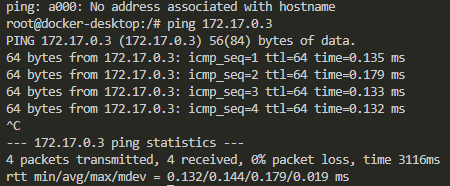
Chúng ta giờ sẽ chuyển sang nhóm user create bridge với tên abc9000

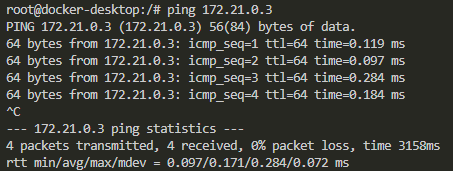
ping được với b001

không thể ping với a001

 cũng không thể ping tới container aaaa

Chúng ta sẽ sử dụng container aaaa để test conect với 4 container kia

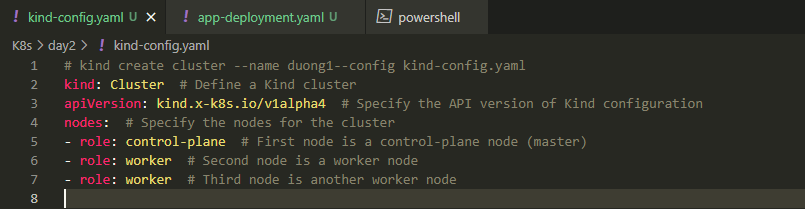
ping tới ip của a001

ping tới ip của b001

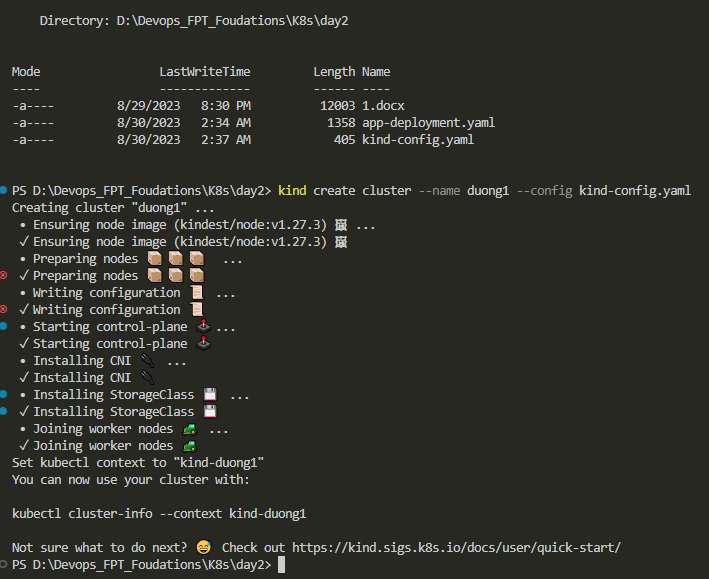
Vậy chúng ta chỉ có thể ping container tại network của chúng với container tại network host thì nó có thể ping tới mọi network còn lại

Task2 : k8s tạo cluster 3 node, 4 pod. Ping các node và các pod với nhau

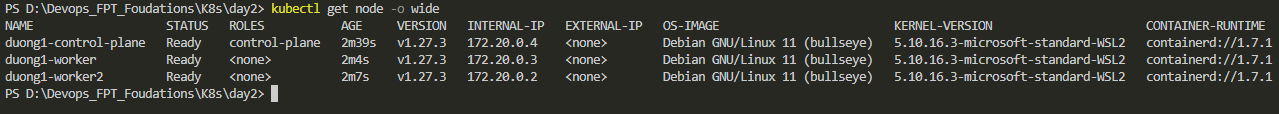
Tạo clusters với file kind-config.yaml ta định nghĩa 3 node 1 node là control-pkane 2 node kia worker



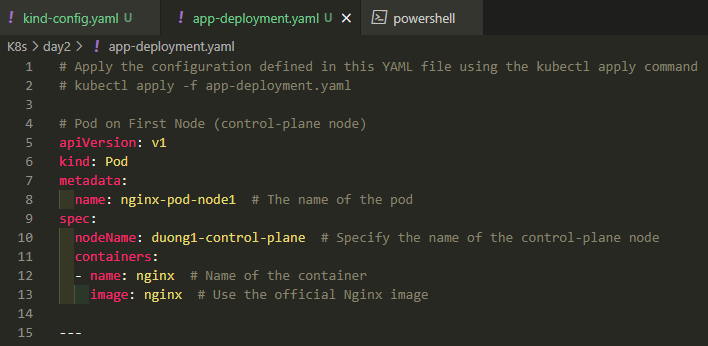
Với tệp tại cùng đường dẫn ta thực hiện lệnh tạo cluster kind create cluster --name duong1 --config kind-config.yaml



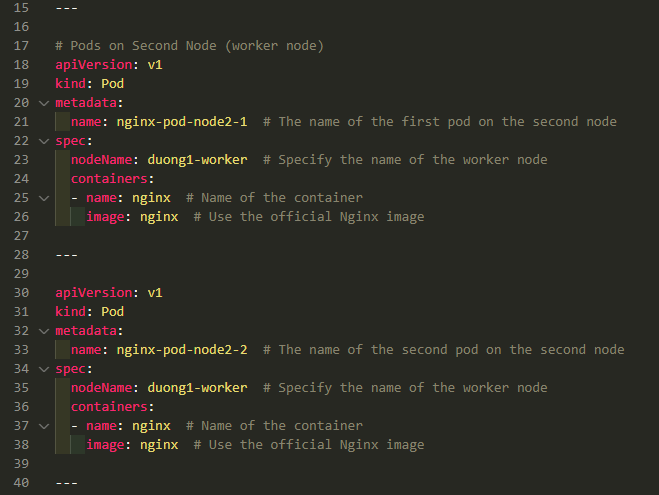
Kiểm tra xem các node mới tạo



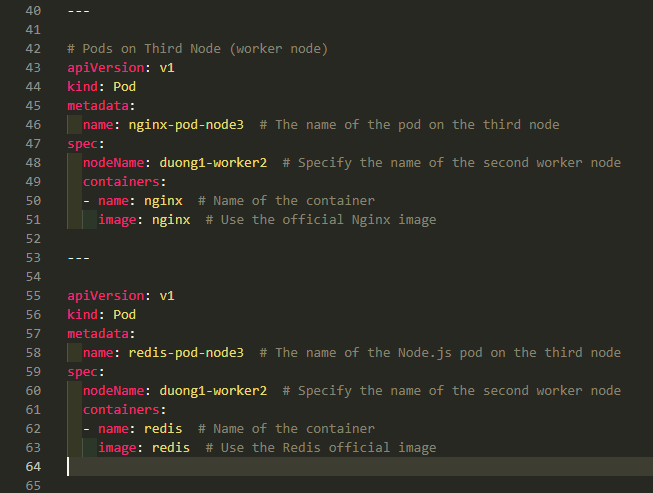
Ta tạo file yaml với ý định tạo cho node 1 control-plane: 1 pod tên nginx-pod-node1: 1 container nginx



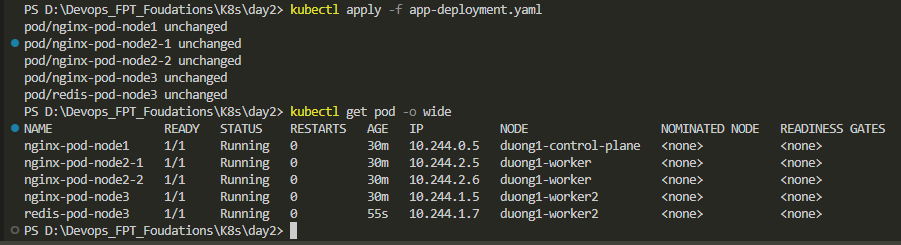
Node 2 sẽ có 2 pod …node2-1 và node2-2 mỗi pod đều mang 1 container nginx



Node 3 có 1 pod tên nginx-pod-node3 có 1 container nginx, 1 pod tên redis có 1 container redis



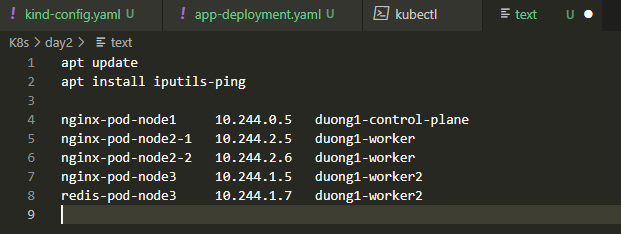
Áp dụng lệnh và chạy kubectl apply -f app-deployment.yaml

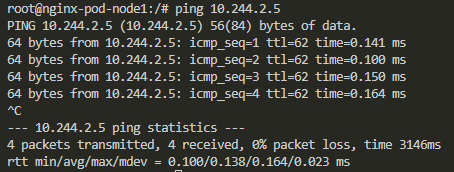


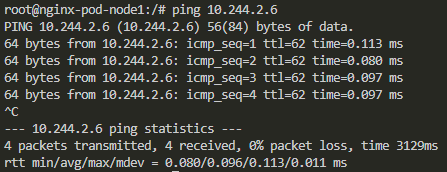
Thứ nhất ta vào container pod 1 cài tool ping theo lệnh

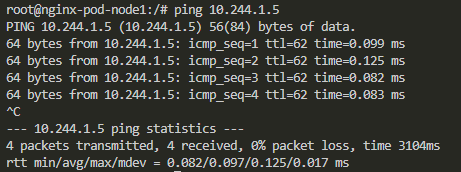
apt update

apt install iputils-ping

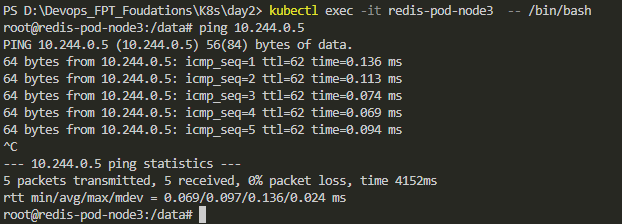
với thông tin sau 

ping tới nginx-node2-1 với ip 10.244.2.5 

ping tới nginx-node2-2 với ip 10.244.2.6 

ping tới nginx-node3 với ip 10.2441.5 

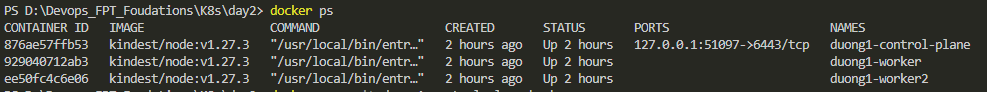
ping tới redis-node3 với ip 10.244.1.7 (vì cổng 80 của redis không mở và không có display ta sẽ vào container redis-node3 cài ping và ping lại nginx-node1)



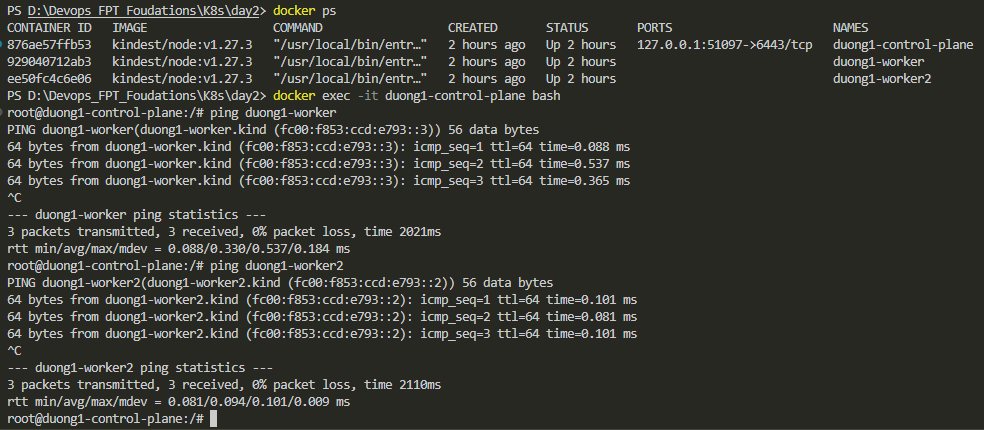
Vậy ta kết luận sau khi thực hành rằng tất cả các pod mặc dù thuộc các node khác nhau nhưng vẫn có thể ping được lại cho nhau (worker to worker and master to worker)

Kiểm tra xem kết nối với các node với nhau

Ta docker ps check các node đã tạo

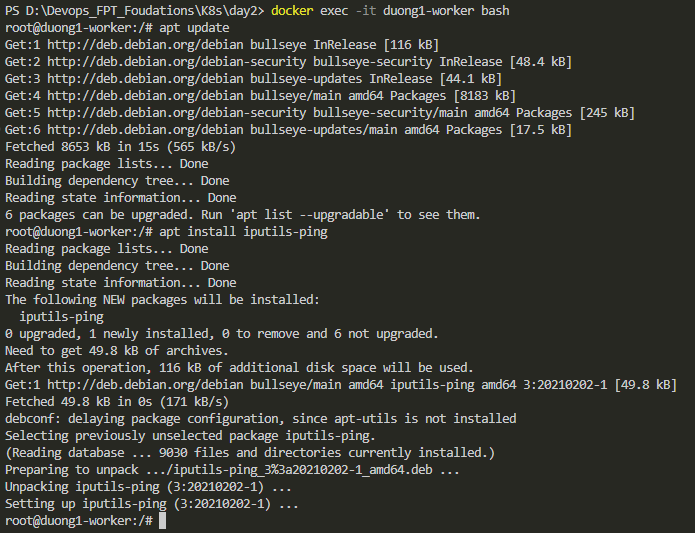


Ta vào trong duong1-control-plane cài ping tool và bắt đầu thử ping 2 worker node kia

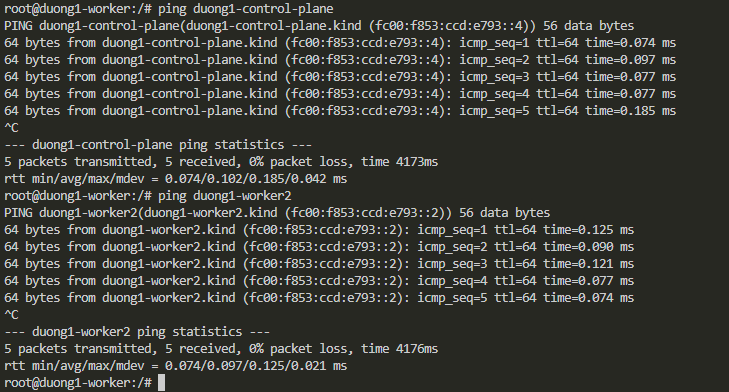


Vậy là đã có thể ping được tới 2 worker node

Ta thử kiểm tra với vị trí là worker node1



Sau khi cài xong ping tool ta thử ping 2 thằng node kia



Vậy sau khi kiểm nghiệm ta kết luận mọi node và mọi pod đều có kết nối được với nhau