// Selis → microservice 150 টা, container প্রায় 5000.

<u>Class 02 : (18 July 2020)</u>

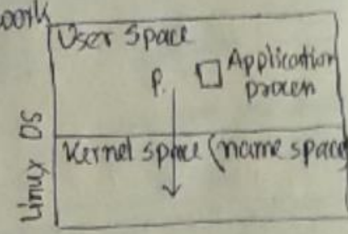
Lab url : Kloudlab.org
// user : sajibeece09@gmail.com
// pass : sajibeece09

// <u>Docker হল product এর নাম।</u>
→ Technology হল: container
// Docker এর document এ আছে :
◻ cgroups = control groups
◻ namespace

<u>Network concept of Docker :</u>

// ours app. থাকে user space-এ.
// ~~kernel space এ থাকে network~~ NIC card call করে
↓
kernel space এ, then ও call করে
↓
user space.

cgroups — resource utilize করে properly



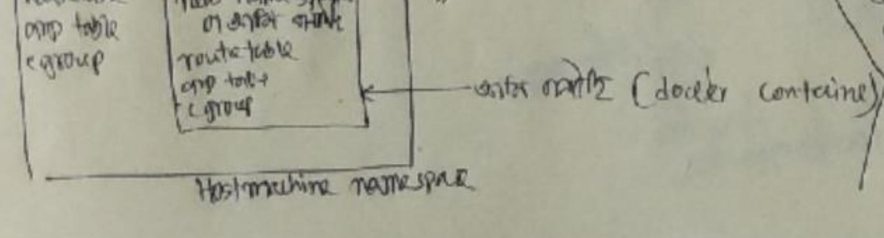
P. এখানে kernel service call করে।

- এই cgroup দিয়েই বলে দেওয়া যায় 50MB এর বেশি RAM নিবে না, 2 core এর বেশি CPU নিবে না
- User space rule বানায়, kernel space এ ঐ rule টা call দিয়ে।

◻ Jenkins এ cgroup থাকলে সুবিধা হত, কারণ build করার সময় machine hunge হয়ে যায়।

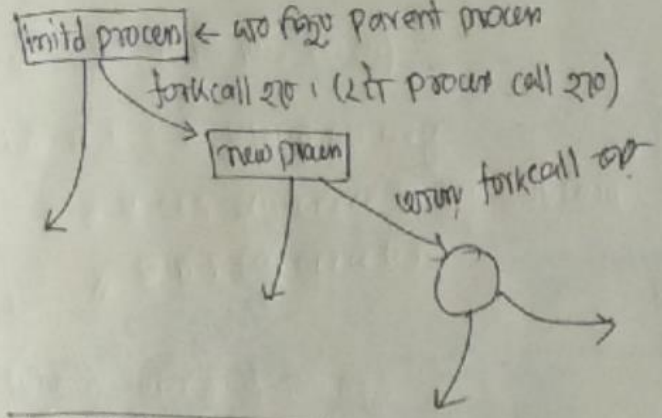
<u>Name Space : (Kernel space)</u> - Docker এর pura concept হচ্ছে এই space এ
- আমাদের নতুন name space তৈরি করে, এই call টা আসে দিয়ে docker এ.

// route table
// arp table



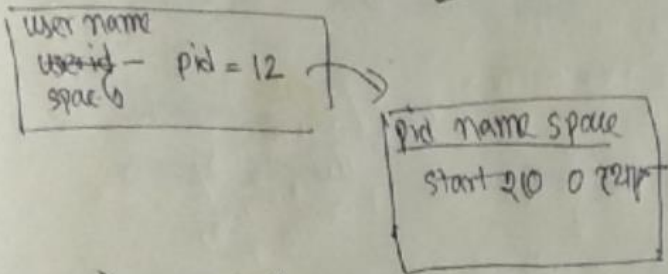
⇒ pid name space
docker launch হলে
pid name space
launch হয়।
process for

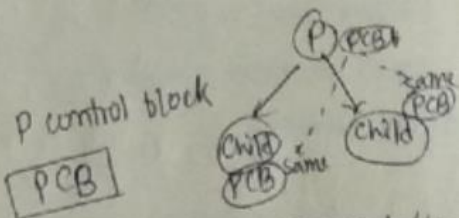
pid name space



// fork call
new process
call

thread
process
difference





process control block
PCB

child PCB
process
replace

→ OS
→ PCB
→ PCB
RAM, CPU, resource info.

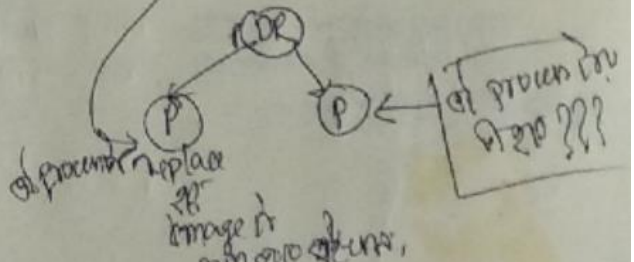
vim
nano

// PCB

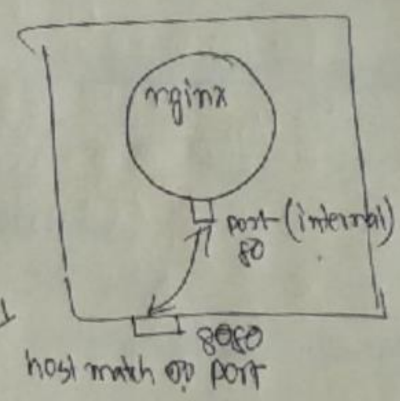
// machine code
save
file system

// Docker build → image

// Docker run → docker daemon
fork call
then
image
process call



// docker run হলো, but আমাদের কে packet পাঠাতে হবে। so, network নিয়ে পড়তে হবে।



labএ যে error port এর conflict clear হয়েছে।

// iptables এ rule সেট হয় port forwarding

// p flag = port forward

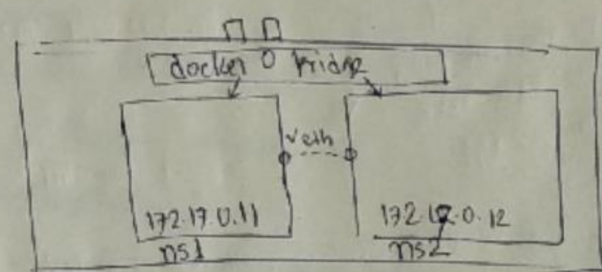
// d flag = deamon অথবা dettach হয়ে যায়
↓
control c দিলেও এর পর off হয় না।

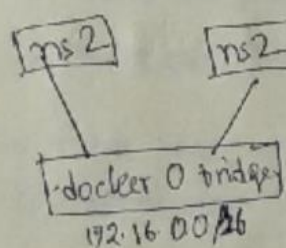
// docker ps
docker ps -a

// docker run -p 8080:80 nginx

-d মানে detach

Virtual ethernet (Veth)
Veth pair (patch cable)



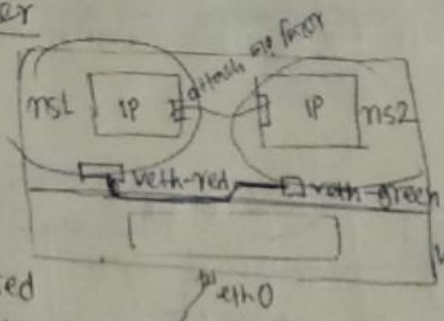


CIDR

// দুইটা name space আছে

IPtable decide করে কোন container এ packet যাবে,
- iptable এ অনেক rule এর chain

Create virtual peer



by default name spaces are isolated.

ip link

Sudo ip netns add red
sudo ip netns add ~~blue~~ green
sudo ip netns exec red ip link

sudo ip netns list now, off name; to connect them both
sudo ip link add veth-red type veth peer name veth-green

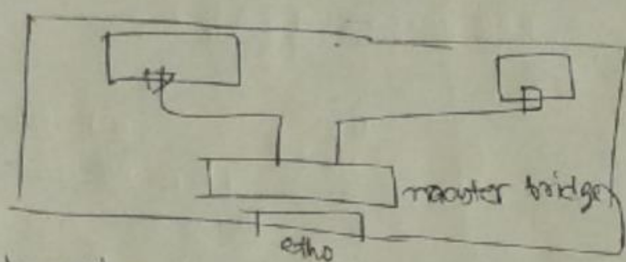
Sudo ip link set veth-red netns red
sudo ip link set veth-green netns green
sudo ip netns exec red ip addr add 192.168.15.1/24 dev veth-red
sudo ip netns exec green ip addr add 192.168.15.2/24 dev veth-green
sudo ip netns exec red ip addr

Sudo ip netns exec red ip link set lo up
sudo ip netns exec red ip link set veth-red up
sudo ip netns exec green ip link set lo up
Sudo ip netns exec green ip link set veth-green up

sudo ip netns exec red ping 192.168.15.2
sudo ip netns exec green ping 192.168.15.1

build your own x repo → ~~docke~~ github (dekho.)

Bocker



sudo ip netns ~~[illegible]~~ exec red arp -a
sudo ip netns exec green arp -a
sudo docker network ls
sudo docker network inspect bridge

[illegible] } network namespace
" docker namespace

A guide to Kubernetes

https//sookocheff.com/pos/kubernetes/understand-kubernetes-networking-mode