Docker Networking

Bridge linkDocker networking poc

Bridge

1. Sudo docker run -itd –rm –name thor busybox
2. Sudo docker run -itd –rm –name mjolnir busybox
3. Sudo docker run -itd –rm –name stromebreaker nginx
4. Bridge link
5. Docker inspect bridge
6. Docker exec -it thor sh (ping each other and internet also in bridge)
7. Sudo docker run -itd –rm -p 80:80 –name stromebreaker nginx(redeploy for accessing the data on host)

User Define Bridge(why?-isolation, docker recommend user define bridge)

8. Sudo docker network create asgard
9. Ip address show
10. Network ls
11. Sudo docker run -itd –rm –network asgard –name loki busybox
12. Sudo docker run -itd –rm –network asgard –name odin busybox

Host Network (By default)-in this you dont need to expose any ip and ports it will work

13. Sudo docker stop stromebreaker
14. Sudo docker run -itd –rm –name –network host stromebreaker nginx(move to near to host)
15. `sudo systemctl restart docker`

Macvlan(conainer directly connect to our physical network-directly connect to our home network)

-o parent = enp6s18 actually tie our macvlan to our host network interface

16. Sudo docker stop thor mjolnir
17. Sudo docker run -itd –rm –network newasgard \
    –ip 192.168.0.61 \
    –name thor busybox
    When you do this then you need to on promiscuous mode
    Our network may not be able to have multiple mac address on one switch port

    How to set promiscuous mode
18. Sudo ip link set enp6s18 promisc on and enable from virtual box networki settings
19. Reboot system

Ipvlan l2, l3

Sudo docker network create -d ipvlan –subnet 192.168.0.0/16 –gateway 192.168.0.1 -o parent=enp6s18 newasgard

Docker run -itd –rm –network newasgard –ip 192.168.0.61 –name thor busybox(same mac as host)

Overlay network - pending (Update soon Regarding this….)

None Network- loopback interface