Docker Networking

Docker networking is one of the most crucial part to understand if you want to be master on docker. The concept is not that much complicated if you put your effort in right direction. It ease your life when you set up your container for talking to another container or expose externally.

Docker engine interacts with Libnetwork which talks to one of the high construct network drivers.

- Native Network Driver
- Native IPAM Driver
- Remote Network Driver
- Remote IPAM Driver

* Docker engine <=> Libnetwork <=> Network drivers

We mostly play with native network driver. Let me brief native network drivers.

- brigde (Default network name: bridge)
- host (Default network name: host)
- null (Default network name: none)
- overlay (Default network name: ingress)
- macvlan (Default network name: macvlan)

Use cases of using above native network drivers and command to create it:

1. BRIDGE: It provides single host networking where all containers on same bridge network on same host can communicate to each other. It acts as virtual switch.(#docker network create -d bridge).
2. HOST: When you need to assign host's IP to container. Container directly listens to on host port. (#docker network create -d host).
3. NULL: When you need to isolate your container from any network. No network communication to container. (#docker network create -d null).
4. OVERLAY: It provides native out-of-box multi-host networking. Generally used in cluster. It uses VXLAN technology and connect VXLAN tunnel between VTEP points on each host. (#docker network create -d overlay).
5. MACVLAN: It allows containers to participate directly in existing networks and VLANs. Promiscuous mode should be set on host's ethernet card. It is least used. (#docker network create -d macvlan)