

Docker Network

Create your own virtual network

Docker Network

- การติดต่อกับ app ที่อยู่ใน container จะติดต่อผ่านทาง network layer เหมือนกับการทำงานกับ virtual machine
- Docker จะสร้าง default network ให้ตั้งแต่ตอนติดตั้ง ทุก container ที่สร้างขึ้นแบบไม่ระบุ network จะอยู่บน default network ทั้งหมด
- เราสามารถ list network ทั้งหมดที่มีบน host ได้ด้วยคำสั่ง

```
$ docker network ls
```

ตัวอย่าง

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
002a55002fd6	bridge	bridge	local
12261b9fcec2	host	host	local
9bc2e9995371	none	null	local

- เราสามารถระบุ network ที่ต้องการตอนสร้าง container ได้ด้วย flag `--network`

Network Drivers

- ที่ version 1.12 Docker รองรับ driver 3 แบบ คือ bridge, host, และ null
 - bridge :- เป็น default network ที่ใช้ติดต่อกับ container ทั้งหมด (ยกเว้น container ที่เราระบุ network) ถ้าติดตั้ง Docker บน Linux แล้วลองใช้คำสั่ง ifconfig เราจะเห็น network ชื่อ docker0
 - none หรือ null :- ใช้สำหรับ container ที่ไม่ต้องการ network interface
 - host :- เป็น network ที่ใช้เฉพาะภายใน host เท่านั้น ซึ่ง configuration ของ network นี้จะเหมือนกับของ host
- เราสามารถดูรายละเอียดของ network ได้ด้วยคำสั่ง inspect ตามด้วยชื่อ network เช่น

```
$ docker network inspect bridge
```
- ถ้าใน network มี container ที่ทำงานอยู่ เมื่อเราสั่งคำสั่ง network inspect เราจะเห็นข้อมูลของ container ใน section “Containers”

Default Network

- เราสามารถเพิ่ม network ใหม่ นอกเหนือจาก default network และกำหนด network ให้ container ได้เมื่อสั่ง run เช่น

```
$ docker run --network=<network_name> <image_id>
```

- เราสามารถลบ network ออกเมื่อไม่ต้องการได้ แต่เราไม่สามารถลบ default network ได้

User-defined Network

- Docker รองรับข้อกำหนด isolation ให้กับกลุ่มของ container ใน default network ได้ผ่านคำสั่ง `docker run --link` แต่เราสามารถสร้าง network ใหม่เพื่อให้กลุ่มของ container ที่เราต้องการแยกออกจาก container อื่นๆ ได้ดีกว่า
- Network ที่ถูกสร้างขึ้นใหม่จะต้องใช้ driver ซึ่งเราสามารถเลือกได้ตั้งแต่ bridge network, overlay network (สำหรับ swarm mode) หรือ MACVLAN network นอกจากนี้ docker ยังมี plugins เพื่อให้นักพัฒนาสร้าง driver ขึ้นมาเองได้อีกด้วย
- เราสามารถสร้าง network ได้มากกว่า 1 ตัว
- container ใน network สามารถสื่อสารกันได้ผ่าน IP Address แต่ไม่สามารถสื่อสารกับ container ที่อยู่ใน network อื่นได้
- container อยู่ในหลาย network ได้

Using Bridge Network

1. สร้าง network ใหม่

```
$ docker network create --driver=bridge --subnet=192.168.0.0/16 mynet
```

note: flag --driver เป็น optional ถ้าไม่ระบุจะได้ bridge network เสมอ

note2: flag --subnet ใช้เพื่อกำหนดกำหนด user subnet ถ้าไม่กำหนดจะได้ default subnet ที่สร้างโดย docker และไม่สามารถกำหนด fix ip เองได้

2. ลอง inspect network ดู

```
$ docker network inspect mine
```

3. สร้าง container ใหม่ โดยระบุว่าใช้ network “mynet”

```
$ docker run -d -t --name myubuntu --network=mynet ubuntu
```

4. ลอง inspect network ดูอีกครั้ง จะเห็นว่ามี container ชื่อ myubuntu อยู่ใน list

Static IP & Subnet

- เราสามารถกำหนด IP ให้กับ container ได้ แต่ทำได้เฉพาะกับ user-defined network เท่านั้น ใช้กับ default network ไม่ได้
- IP address จะต้องอยู่ใน subnet เดียวกับ user-defined network เท่านั้น (inspect network ดู)

```
$ docker run -dt --network=mynet --ip 192.168.0.100 ubuntu
```

note: ตอนสร้าง network ต้องกำหนด subnet จึงจะกำหนด fix ip ได้

- นอกจากนั้น ยังสามารถกำหนด hostname ให้กับ container ได้ด้วย (ใช้กับ default network ได้)

```
$ docker run -dt --hostname ubuntu1 ubuntu
```

- เราสามารถกำหนด subnet ให้กับ network ได้

```
$ docker network create --subnet=172.10.0.1/16 mynet
```

Connect Container to Network

- เราสามารถเพิ่ม (attach) container ที่มีอยู่แล้ว เข้าไปใน network ที่ต้องการได้

```
$ docker network connect <network_name> <container_id>
```

เช่น

```
$ docker run -dt --name myubuntu ubuntu
```

```
$ docker network connect mynet myubuntu
```

- ถ้าต้องการเอา container ออกจาก network ใช้ parameter “disconnect”

```
$ docker network disconnect <network_name> <container_id>
```

เช่น

```
$ docker network disconnect bridge myubuntu
```


Overlay Driver

- ในกรณีที่ container ทำงานกับ swarm mode และเราต้องการให้ container แยกอยู่คนละ network เราต้องกำหนดประเภทของ driver เป็นแบบ overlay
- container ที่ทำงานด้วย docker run จะไม่สามารถใช้ network แบบ overlay ได้
- ตัวอย่าง

```
$ docker network create \  
  --driver overlay \  
  --subnet 10.0.9.0/16 \  
  my-multihost-network
```

```
$ docker service create --replica 3 \  
  --network my-multihost-network \  
  --name myweb \  
  nginx
```