docker0 ip 변경

Docker

개요

- docker는 기본브릿지 docker0생성
- docker0 인터페이스의 기본 ip는 172.17.0.0/16
- 이는 해당 host에 올라가는 container의 IP영역을 결정
- docker 설치전 기존 vm 또는 서버에서 어떠한 이유로 172.17.0.0 대역을 사용 중이라면 docker 서비스 시작 후 기존 172.17.0.0 대역은 통신이 끊어지고 docker로 대체

docker0 ip 변경

✓ Docker version

```
$ docker version
Client:
 Version:           18.09.3
 API version:       1.39
 Go version:        go1.10.8
 Git commit:        774a1f4
 Built:             Thu Feb 28 06:53:11 2019
 OS/Arch:           linux/amd64
 Experimental:      false
```

1. docker service stop

```
$ sudo systemctl stop docker


$ sudo systemctl status docker
 docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2019-07-30 00:18:19 UTC; 1min 6s ago
     Docs: https://docs.docker.com
  Process: 962 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
(code=exited, status=0/SUCCESS)
 Main PID: 962 (code=exited, status=0/SUCCESS)

Jul 30 00:14:13 edu dockerd[962]: time="2019-07-30T00:14:13.826089834Z" level=warning msg="Error (Unable
to complete atomic operation, key modified) deleting object [endpoint c129840d6a2684d418b4bab5628
Jul 30 00:14:13 edu dockerd[962]: time="2019-07-30T00:14:13.911258717Z" level=info msg="Default bridge
(docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a pr
Jul 30 00:14:13 edu dockerd[962]: time="2019-07-30T00:14:13.955643013Z" level=info msg="Loading
containers: done."
Jul 30 00:14:14 edu dockerd[962]: time="2019-07-30T00:14:14.217215684Z" level=info msg="Docker daemon"
commit=774a1f4 graphdriver(s)=overlay2 version=18.09.3
Jul 30 00:14:14 edu dockerd[962]: time="2019-07-30T00:14:14.218085698Z" level=info msg="Daemon has
completed initialization"
Jul 30 00:14:14 edu systemd[1]: Started Docker Application Container Engine.
Jul 30 00:14:14 edu dockerd[962]: time="2019-07-30T00:14:14.273713499Z" level=info msg="API listen on
/var/run/docker.sock"
Jul 30 00:18:19 edu systemd[1]: Stopping Docker Application Container Engine...
Jul 30 00:18:19 edu dockerd[962]: time="2019-07-30T00:18:19.155336961Z" level=info msg="Processing
signal 'terminated'"
Jul 30 00:18:19 edu systemd[1]: Stopped Docker Application Container Engine.
```

2. docker service stop 해도 docker0은 남아있음을 확인

```
$ ifconfig
br-c129840d6a26: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
        ether 02:42:c9:8e:8c:9b  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:57:56:96:11  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fead:5c4f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ad:5c:4f  txqueuelen 1000  (Ethernet)
        RX packets 4162  bytes 5066663 (5.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1004  bytes 102910 (102.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 108  bytes 26475 (26.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 108  bytes 26475 (26.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

✓ **우선순위**

ifconfig에서 나오는 interface의 우선순위(나오는 순서)는 변경 불가

3. docker interface 상태 정보 확인

```
$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default
qlen 1000
    link/ether 08:00:27:ad:5c:4f brd ff:ff:ff:ff:ff:ff
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group
default
    link/ether 02:42:57:56:96:11 brd ff:ff:ff:ff:ff:ff
4: br-c129840d6a26: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT
group default
    link/ether 02:42:c9:8e:8c:9b brd ff:ff:ff:ff:ff:ff
```

4. docker ip addr

```
$ ip addr | grep docker0
3: docker0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
```

ip addr

항목	설명
LOOPBACK	자신의 호스트로 보내는 인터페이스 의미(=localhost)
BROADCAST	브로드캐스트 패킷을 처리할 수 있음을 의미, 해당 기능을 통해 DHCP 서버로부터 IP주소를 받을 수 있음
MULTICAST	멀티캐스트 패킷을 처리할 수 있음
UP	해당 NIC가 작동중임
LOWER_UP	L1 레이어 즉, 물리계층에서 신호가 UP이라는 의미
mtu	해당 프로토콜이 해당 레이어에서 전송할 수 있는 최대의 단위(바이트)로 이더넷의 기본값은 1500으로 설정
qdisc	Queuing Disciplines를 의미하며 NIC에 들어오기전에 데이터 패킷들이 Queue에 저장되는데 FIFO형식인 Queue에 넣기전에 패킷에 우선순위를 부여해서 스케쥴링하는 부분으로 현재 eth0의 경우 pfifo_fast라는 방식을 사용한다는 의미
state	현재 NIC의 작동상태를 의미
group	인터페이스 그룹. 기본값은 default
qlen	전송큐의 크기
link/ether	L2 레이어 즉, Link Layer의 프로토콜이 Ethernet이라는 의미이며 바로 옆에 나오는 주소는 해당 NIC의 MAC주소이고 brd는 브로드캐스트를 할 때의 주소
inet	L3 레이어 즉, Network Layer가 인터넷임을 의미하며 바로 옆에 나오는 주소는 ipv4와 ipv6에 따른 주소를 의미
scope	해당 인터페이스가 어느 수준에서 접근가능하며 유효하냐는 의미입니다. Global의 경우 외부 네트워크에서 접근이 가능한 범위이며(클라우드에서 호스팅한 인스턴스에 들어가면 Global로 나와있는 경우가 있습니다.) Link의 경우 현재 인터페이스가 속한 LAN안에서만 접근이 가능하고 유효하며 Host의 경우는 현재 인터페이스가 속해있는 호스트에서만 유효하고 접근이 가능하다
valid_lft, preferred_lft	자료의 내용을 참고하면 valid_lft(Valid Lifetime)은 해당 주소가 유효한 시간을 의미하며 뒤에 나온 preferred_lft(Preferred Lifetime)보다 크거나 같다고 합니다. preferred_lft은 의미 그대로 해당 주소가 유효했으면 하는 설정값입니다.

출처: https://twpower.github.io/146-brief-explanation-about-ip-address

5. docker config open

```
$ sudo vi /etc/docker/daemon.json
```

6. 21번째 줄에 아래 내용 추가 (8번에서 설정한 ip) → vi 저장 후 종료

docker

```
{
  "bip": "192.168.100.1/24"
}
```

7. docker 시작

```
$ sudo systemctl start docker


$ sudo systemctl status docker
 docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-07-30 02:24:23 UTC; 13s ago
     Docs: https://docs.docker.com
 Main PID: 4650 (dockerd)
    Tasks: 10
   CGroup: /system.slice/docker.service
           4650 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.282060600Z" level=warning msg="Your kernel
does not support swap memory limit"
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.282104385Z" level=warning msg="Your kernel
does not support cgroup rt period"
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.282117988Z" level=warning msg="Your kernel
does not support cgroup rt runtime"
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.282614143Z" level=info msg="Loading
containers: start."
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.533262802Z" level=info msg="Default bridge
(docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a p
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.571424853Z" level=info msg="Loading
containers: done."
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.621083153Z" level=info msg="Docker daemon"
commit=774a1f4 graphdriver(s)=overlay2 version=18.09.3
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.621170931Z" level=info msg="Daemon has
completed initialization"
Jul 30 02:24:23 edu dockerd[4650]: time="2019-07-30T02:24:23.641746998Z" level=info msg="API listen on
/var/run/docker.sock"
Jul 30 02:24:23 edu systemd[1]: Started Docker Application Container Engine.
```

8. **ifconfig 확인**

```
$ ifconfig
br-c129840d6a26: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
        ether 02:42:c9:8e:8c:9b  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.100.1  netmask 255.255.255.0  broadcast 192.168.100.255
        ether 02:42:57:56:96:11  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fead:5c4f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ad:5c:4f  txqueuelen 1000  (Ethernet)
        RX packets 7983  bytes 5407813 (5.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3309  bytes 519619 (519.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1343  bytes 413411 (413.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1343  bytes 413411 (413.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

docker-compose

개요

- docker0을 변경하여도 docker-compose로 컨테이너를 올리면 새로운 브리지가 네트워크 interface가 생성됨 (172.17.0.1 또는 172.18.0.1)
- docker-compose로 컨테이너를 올리려면 별도의 설정을 해줘야함

HFN의 BYFN을 docker-compose up 시킬 경우(docker0 ip를 변경했을 경우)

1. network up

```
$ ./byfn up
```

2. docker network list (byfn으로 생성된 network bridge는 net_byfn)

```
$ docker network list
NETWORK ID          NAME                DRIVER              SCOPE
f5d7eab82498        bridge              bridge              local
65922d52cb60        host                host                local
c129840d6a26        net_basic           bridge              local
3f8a8a7ec0f3        net_byfn            bridge              local
2147d7369089        none                null                local
```

3. net_byfn network inspect

```
$ docker network inspect 3f8a8a7ec0f3
[
    {
        "Name": "net_byfn",
        "Id": "3f8a8a7ec0f354f9b31f63dfe088fb86b1c517e72e228b1b234ac2b86756033b",
        "Created": "2019-07-30T04:42:11.714969596Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "2a8f6d84c6589c74c5f326606b1605262f62240a470d7f1d9132af670102de18": {
                "Name": "peer0.org1.example.com",
                "EndpointID": "c729fd17a1972b9f3c3e830098037be68dd521a68be9836a7325c694f3e719a9",
                "MacAddress": "02:42:ac:11:00:04",
                "IPv4Address": "172.17.0.4/16",
                "IPv6Address": ""
            },
            "2d40192bb00935a9bc4ed0a84b442082236a2cb013776052e7c40c82c4d5705d": {
                "Name": "dev-peer0.org1.example.com-mycc-1.0",
                "EndpointID": "acb4910c606335b2c574e85dce16d44e7783b46e39079057d696ac5435e27190",
                "MacAddress": "02:42:ac:11:00:09",
                "IPv4Address": "172.17.0.9/16",
                "IPv6Address": ""
            },
            "458c7df997179d33d6db8330d8d7ef90809eaf037cc53a81e35841f79ed3105c": {
                "Name": "dev-peer0.org2.example.com-mycc-1.0",
                "EndpointID": "808b8366da2171fe4cda07b501230738b26935f37825fdd385a9768c479d55e3",
                "MacAddress": "02:42:ac:11:00:08",
                "IPv4Address": "172.17.0.8/16",
                "IPv6Address": ""
            },
            "5d3ab1f1f8ee3c121a27ef0ec902dfa7fa1016841084d3eb61bf127081ebf08b": {
                "Name": "cli",
                "EndpointID": "84e4abb224e268b4f575c96dc7f09e45153b2d3f58880a9099e93858e4ced7a8",
                "MacAddress": "02:42:ac:11:00:07",
                "IPv4Address": "172.17.0.7/16",
                "IPv6Address": ""
            },
            "5db4a81235aaa8652cc1c0afb9ef0dec1400e1e819ad4c857e3a8e804124d8b3": {
                "Name": "peer1.org1.example.com",
                "EndpointID": "e4e2ae30aa707168be1bf859816bee0d0536dea170b0639fd3c45676aa1d9fb2",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            },
            "753027735b948dce718e868832a798384e9162ea1d003ad5e59df907191997aa": {
                "Name": "peer0.org2.example.com",
                "EndpointID": "ef4c2447789169464d92011cbd719002b6659d2806eda475b3baf5049207d7f1",
                "MacAddress": "02:42:ac:11:00:06",
                "IPv4Address": "172.17.0.6/16",
                "IPv6Address": ""
            },
            "7a900736cb6a219307b06bde733037e56dcaea4e6d3aaa1c19d7baf688785016": {
                "Name": "orderer.example.com",
                "EndpointID": "0b7aa5b2180beab9a9e3e4fd878559a7c1d85f35ba645ab3dc42173335f15846",
                "MacAddress": "02:42:ac:11:00:05",
                "IPv4Address": "172.17.0.5/16",
                "IPv6Address": ""
            },
            "87a03c3f1bfbf4ab63e804f8563c61c1383b52c735e96f5b50b6d58edda93989": {
                "Name": "dev-peer1.org2.example.com-mycc-1.0",
                "EndpointID": "57ebb68bcc5c3d84d6e7251dea92c7eafdf17aa16ec1eb7178d3a78cf312461f",
                "MacAddress": "02:42:ac:11:00:0a",
                "IPv4Address": "172.17.0.10/16",
                "IPv6Address": ""
            },
            "93fbc31d96b8fd98d147af1a8fb90870ca4ecdd35bce5bdd507c5fc61d7c7dc5": {
                "Name": "peer1.org2.example.com",
                "EndpointID": "eb9d8898b851d85d4d57319dcf05b998f5ab91ef17da621e035640f378ff7486",
                "MacAddress": "02:42:ac:11:00:03",
                "IPv4Address": "172.17.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

확인해 보면 Container에 모든 ip가 위에서 생성한 ip 192 대역이 아니라 172대역으로 생성됨을 확인할 수 있음

4. ifconfig 확인

```
$ ifconfig
br-3f8a8a7ec0f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:d7ff:fefe:4cc5  prefixlen 64  scopeid 0x20<link>
        ether 02:42:d7:fe:4c:c5  txqueuelen 0  (Ethernet)
        RX packets 21  bytes 588 (588.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 1416 (1.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

* docker0 말고 새로운 172.17.0.1 브릿지가 생성됨(br-3f8a8a7ec0f3)

5. brctl show

```
$ brctl show
bridge name     bridge id               STP enabled     interfaces
br-3f8a8a7ec0f3         8000.0242d7fe4cc5       no              veth0e7b675
                                                        veth280c9fb
                                                        veth3b28f2e
                                                        veth5eed3c4
                                                        veth6a7af67
                                                        veth8867b98
                                                        veth8c9636c
                                                        veth9246967
                                                        vethacd6ae3
br-c129840d6a26         8000.0242c98e8c9b       no
docker0         8000.024257569611       no              veth1d5d610
```

* docker network inspect 결과와 마찬가지로 네트워크 브리지를 확인해보면 container들이 docker0에 물려있는게 아니라 br-3f8a8a7ec0f3에 물려있음을 확인할 수 있음

6. docker compose를 이용하지 않고 docker image를 pull 받은뒤 run 시키면 docker0 으로 붙음
7. 따라서 docker compose로 컨테이너를 up 시킬때 ip를 변경하려면 docker-compoose 설정 변경해야함

docker-compose up ip 변경(docker0 에서 변경한 ip 쓰면 안됨)

1. **docker compose 파일의 network 부분 수정**
   - 수정 전

     docker-compose-cli.yaml

     ```
     networks:
       byfn:
     ```

   - 수정 후

     docker-compose-cli.yaml

     ```
     networks:
       byfn:
         ipam:
           driver: default
           config:
             - subnet: 192.168.100.1/24
     ```

     위에서 설정한 docker0 ip 변경한 값과 같게 subnet을 넣으면 docker-compose up 할때 다음과 같은 에러 남

     ```
     ERROR: Pool overlaps with other one on this address space
     ERROR !!!! Unable to start network
     ```

2. **docker network inspect 및 brctl show 명령으를 통해 확인해보면 변경한 192.168.200. 0 대역으로 잘 물려있음**

3. **docker compose up 하면 ifconfig 에 새로운 network interface가 생성됨**

4. **기존에 docker0 은 docker network 중 default bridge(name:bridge)에 할당되며**

5. **HFN 같은 경우 기본 bridge 를 사용안하고 network를 추가 하므로 network interface를 새로 생성함**

참고 사이트

- docker network
  - http://linux.systemv.pe.kr/docker-%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EC%9D%B4%ED%95%B4/
- docker ip 변경
  - https://blog.neonkid.xyz/87
  - https://bluese05.tistory.com/16
  - https://thisstory.tistory.com/entry/Docker0-default-network-IP-address-%EB%B3%80%EA%B2%BD-docker-%EB%B8%8C%EB%A6%BF%EC%A7%80-%EA%B8%B0%EB%B3%B8IP-%EC%A3%BC%EC%86%8C-%EB%B3%80%EA%B2%BD
- docker-compose ip 변경
  - https://support.getjoan.com/hc/en-us/articles/360000155325-Change-the-Docker-compose-default-IP-address
  - https://mogutan.wordpress.com/2016/12/28/change-docker-network-bridge-ip-range/