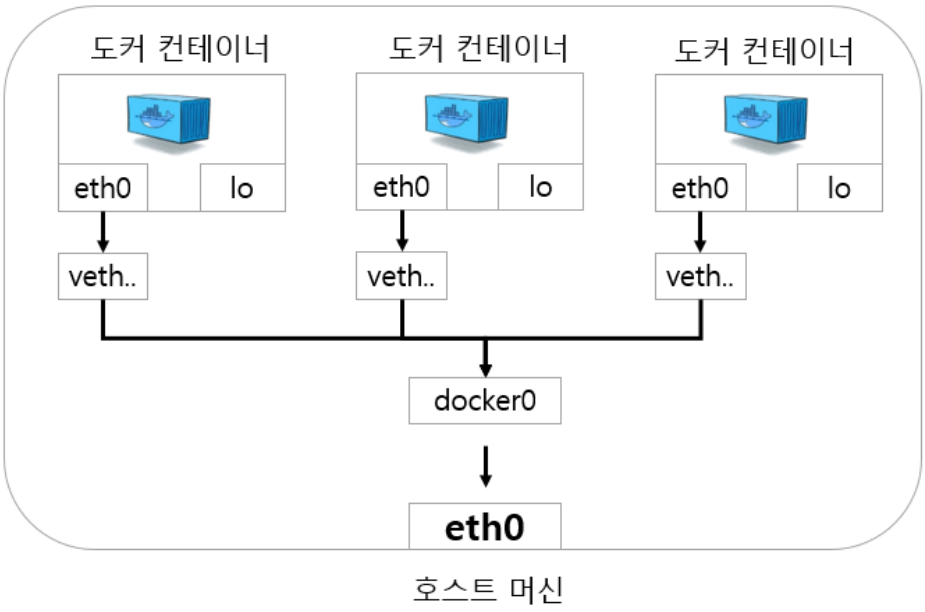
Docker2

도커 네트워크

도커 네크워크 구조

도커는 컨테이너 내부 IP를 순차적으로 할당하며, 이 IP는 컨테이너를 재시작할 때마다 변경될 수 있다. 내부 IP는 도커가 설치된 호스트, 즉 내부 망에서만 쓸 수 있는 IP이므로 외부와 연결이 필요하다. 이 과정은 함으로써 이뤄진다. 도커는 각 컨테이너에 외부와의 네트워크를 제공하기 위해 컨테이너를 시작할 때마다 호스트에 veth...라는 가상 네트워크 인터페이스를 생성한다.

docker0이라는 브리지도 존재하는데 docker0 브리지는 각 veth 인터페이스와 바인딩돼 호스트의 eth0 인터페이스와 이어주는 역할을 한다. 즉 컨테이너의 eth0 인터페이스는 호스트의 veth...라는 인터페이스와 연결이 되었으며 veth 인터페이스는 docker0 브리지와 바인딩돼 외부와 통신할 수 있다.



도커 네트워크 기능

컨테이너를 생성하면 기본적으로 docker0 브리지를 통해 외부와 통신할 수 있는 환경을 사용할 수 있지만 사용자의 선택에 따라 여러 네트워크 드라이버를 쓸 수 있다. 도커가 자체적으로 제공하는 대표적인 네트워크 드라이버는 브리지, 호스트, 논, 컨테이너, 오버레이가 있다.

브리지 네트워크

- 컨테이너를 생성할 때 자동으로 연결되는 docker0 브리지를 활용하도록 설정됨
- 이 네트워크는 172.17.0.x IP 대역을 컨테이너에 순차적으로 할당
- 기본적으로 존재하는 docker0을 사용하는 브리지 네트워크가 아닌 새로운 브리지 타입의 네트워크를 생성할 수도 있음

호스트 네트워크

- 네트워크를 호스트로 설정하면 호스트의 네트워크 환경을 그대로 쓸 수 있음
- 브리지 드라이버 네트워크와 달리 호스트 드라이버의 네트워크는 별도로 생성할 필요 없이 기존의 host라는 이름의 네트워크를 사용
- 컨테이너의 네트워크를 호스트 모드로 설정하면 컨테이너 내부의 애플리케이션을 별도의 포트 포워딩 없이 바로 서비스할 수 있음

논 네트워크

- 아무런 네트워크를 쓰지 않고, 외부와 단절

컨테이너 네트워크

- 다른 컨테이너의 네트워크 네임스페이스 환경을 공유할 수 있음. 공유되는 속성은 내부 IP, 네트워크 인터페이스의 맥주소 등.
- 다른 컨테이너의 네트워크 환경을 공유하면 내부 IP를 새로 할당받지 않으며 호스트에 weth로 시작하는 가상 네트워크 인터페이스도 생성되지 않음

도커스웜

하나의 호스트 머신에서 도커 엔진을 구동하다가 CPU나 메모리, 디스크 용량과 같은 자원이 부족한 경우 가장 많이 사용하는 방법은 여러대의 서버를 클러스터로 만들어 자원을 병렬로 확장. 도커에서 공식적으로 제공하는 도커 스웜과 스웜 모드

스웜 클래식과 스웜 모드는 여러 대의 도커 서버를 하나의 클러스터로 만들어 컨테이너를 생성하는 여러 기능을 제공. 다양한 전략을 세워 컨테이너를 특정 도커 서버에 할당할 수 있고 유동적으로 서버를 확장할 수도 있음. 또한 스웜 클러스터에 등록된 서버의 컨테이너를 쉽게 관리할 수 있음.

도커스웜의 두가지 종류

1. 컨테이너로서의 스웜(스웜 클래식)
    - 여러 대의 도커 서버를 하나의 지점에서 사용하도록 단일 접근점을 제공
    - docker run, docker ps 등 일반적인 도커 명령어와 도커 API로 클러스의 서버를 제어하고 관리할 수 있는 기능을 제공
    - 각종 정보를 저장하고 동기화하는 분산 코디네이터, 클러스터 내의 서버를 관리하고 제어하는 매니저, 에이전트 등이 별도로 실행되어야함
2. 스웜모드
    - 마이크로서비스 아키텍처의 컨테이너를 다루기 위한 클러스링 기능에 초점을 맞춤
    - 같은 컨테이너를 동시에 여러개 생성해 필요에 따라 유동적으로 컨테이너 수를 조절할 수 있고, 컨테이너로의 연결을 분산하는 로드밸런싱 기능을 자체적으로 지원
    - 서비스 확장성과 장정성 측면에서 스웜 클래식보다 뛰어나기 때문에 일반적으로 더 많이 사용
    - 클러스터링을 위한 모든 도구가 도커 엔진 자체에 내장돼 쉽게 서버 클러스터를 구축할 수 있음
    - 마이크로서비스 아키텍쳐 애플리케이션을 컨테이너로 구축할 수 있는 것을 도와줄 뿐 아니라, 서비스 장애에 대비한 고가용성과 부하 분산을 위한 로드밸런싱 기능을 제공하기 때문에 대규모 클러스터에서 서비스를 운영하는 것을 계획하고 있으면 스웜모드를 사용하는 게 더 좋음

도커 스웜 모드의 구조

- 워커 노드 : 실제로 컨테이너가 생성되고 관리되는 도커 서버
- 매니저 노드 : 워커 노드를 관리하기 위한 도커 서버. 매니저 노드는 컨테이너가 생성될 수 있어 워커 노드의 역할을 포함
- 스웜 모드는 매니저 노드와 워커 노드로 구성. 매니저 노드는 1개 이상 있어야하지만 워커노드는 없을 수도 있음.
- 매니저의 부하를 분산하고 특정 매니저 노드가 다운됐을 때 정상적으로 스웜 클러스터를 유지하기 위해 실제 운영환경에서 스웜 모드로 도커 클러스터를 구성하려면 매니저 노드를 다중화하는 것을 권장.

- 스웜 모드는 매니저 노드의 절반 이상에 장애가 생겨 정상적으로 작동하지 못할 경우 장애가 생긴 매니저 노드가 복구될 때까지 클러스터의 운영을 중단. 따라서 가능한 홀수 새로 구성하는 것을 권장

도커 컴포즈

여러 개의 컨테이너가 하나의 애플리케이션으로 동작할 때 매번 run 명령어에 옵션을 설정해 CLI로 컨테이너를 생성하기보다는 여러 개의 컨테이너를 하나의 서비스로 정의해 컨테이너 묶음으로 관리하기 위해 사용. 도커 컴포즈는 컨테이너를 이용한 서비스의 개발과 CI를 위해 여러개의 컨테이너를 하나의 프로젝트로서 다룰 수 있는 작업 환경을 제공

- 도커 컴포즈는 여러 개의 컨테이너의 옵션과 환경을 정의한 파일을 읽어 컨테이너를 순차적으로 생성하는 방식으로 동작
- 도커 컴포즈의 설정 파일은 run 명령어의 옵션을 그대로 사용할 수 있으며, 각 컨테이너의 의존성, 네트워크, 볼륨 등을 정의할 수 있음
- 스웜 모드의 서비스와 유사하게 설정 파일에 정의된 서비스의 컨테이너 수를 유동적으로 조절할 수 있으면 컨테이너의 서비스 디스커버리로 자동으로 이뤄짐
- 도커 컴포즈는 컨테이너의 설정이 정의된 YAML파일을 읽어 도커 엔진을 통해 컨테이너를 생성

YAML 파일의 작성

- 버전 정의, 서비스 정의, 볼륨 정의, 네트워크 정의의 4가지 항목으로 구성
- 가장 많이 사용하는 것은 서비스 정의이며, 볼륨 정의와 네트워크 정의는 서비스로 생성된 컨테이너에 선택적으로 사용
- 각 항목의 하위 항목을 정의하려면 2개의 공백(스페이스)으로 들여쓰기 해서 상위 항목과 구분
- YAML 파일에 네트워크 항목을 정의하지 않으면 도커 컴포즈는 프로젝트별로 브리지 타입의 네트워크를 생성

참고자료 : <도커/쿠버네티스>