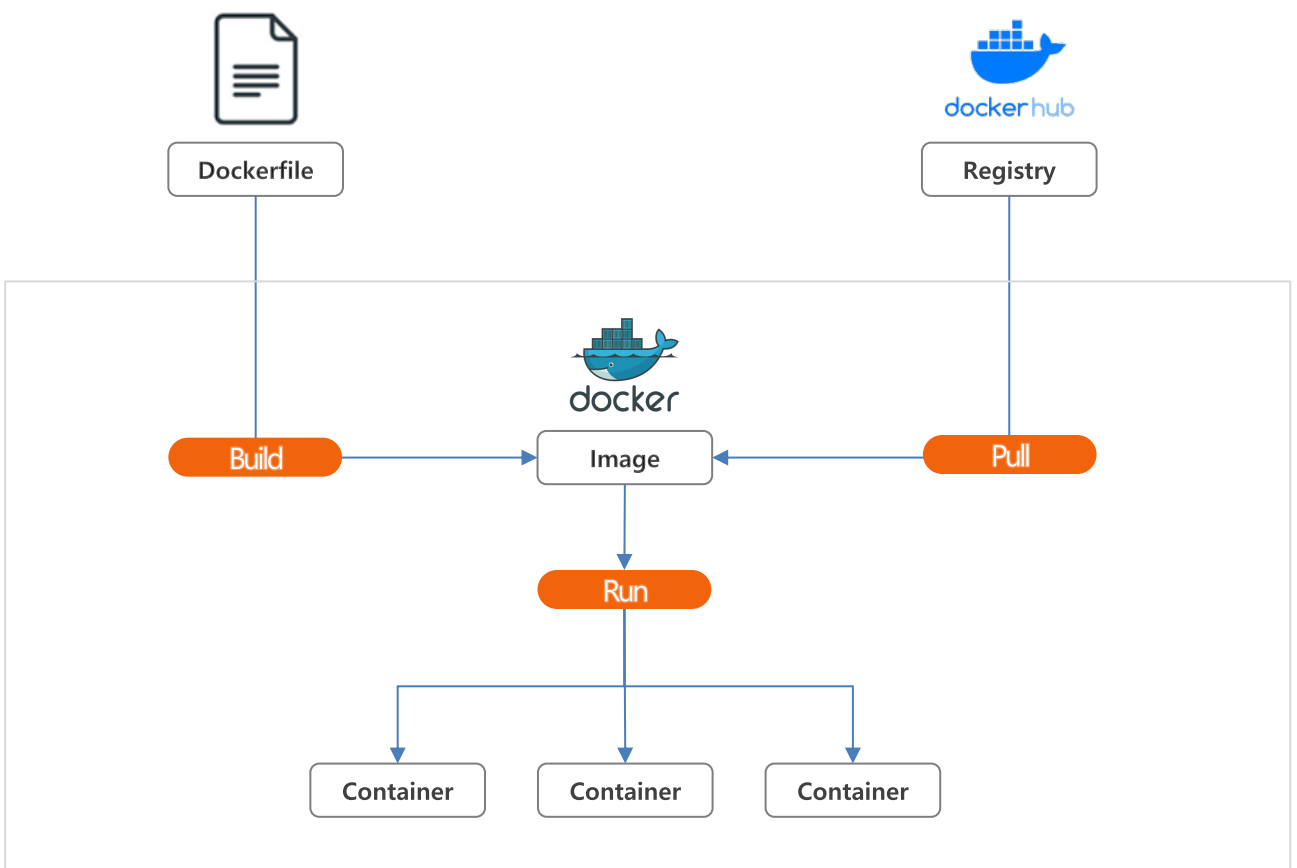


1. Docker 의 개념

- 소프트웨어 개발과 배포를 위한 플랫폼으로, 컨테이너라는 기술을 사용하여 응용 프로그램을 포장하고 실행

2. Docker 구성 요소

- **도커 파일(Dockerfile)** : 도커 이미지를 빌드하기 위한 스크립트입니다. 응용 프로그램과 그 의존성을 포함하는 이미지를 어떻게 구축할지 지정합니다.
- **이미지(Image)** : 응용 프로그램을 실행하는 데 필요한 모든 것을 포함하는 경량, 독립적, 실행 가능한 소프트웨어 패키지입니다.
- **레지스트리(Registry)** : 도커 이미지를 저장하고 공유하는 서비스입니다. (ex : Docker Hub)
- **컨테이너(Container)** : 도커 컨테이너는 이미지의 실행 인스턴스입니다. 컨테이너는 격리된 환경에서 실행되며, 자체 파일 시스템, 네트워킹, 독립적인 프로세스 ID 공간을 갖습니다.
- **볼륨(Volume)** : 데이터를 저장하고 컨테이너 간에 데이터를 공유하기 위한 방법입니다. 컨테이너의 볼륨은 컨테이너가 삭제되는 경우, 같이 삭제됩니다.
- **네트워크(Network)** : 기본적으로 가상 네트워크 인터페이스 안에서 컨테이너 간 통신이 이루어 집니다. 다양한 네트워크 드라이버(예: bridge, overlay, host)를 통해 컨테이너가 서로 또는 외부와 통신할 수 있게 합니다.
- **도커 컴포즈(Docker Compose)** : 여러 컨테이너로 구성된 도커 애플리케이션의 서비스, 네트워크, 볼륨 등을 정의하고 실행하기 위한 도구입니다.



1. 공개 IP (Public IP)

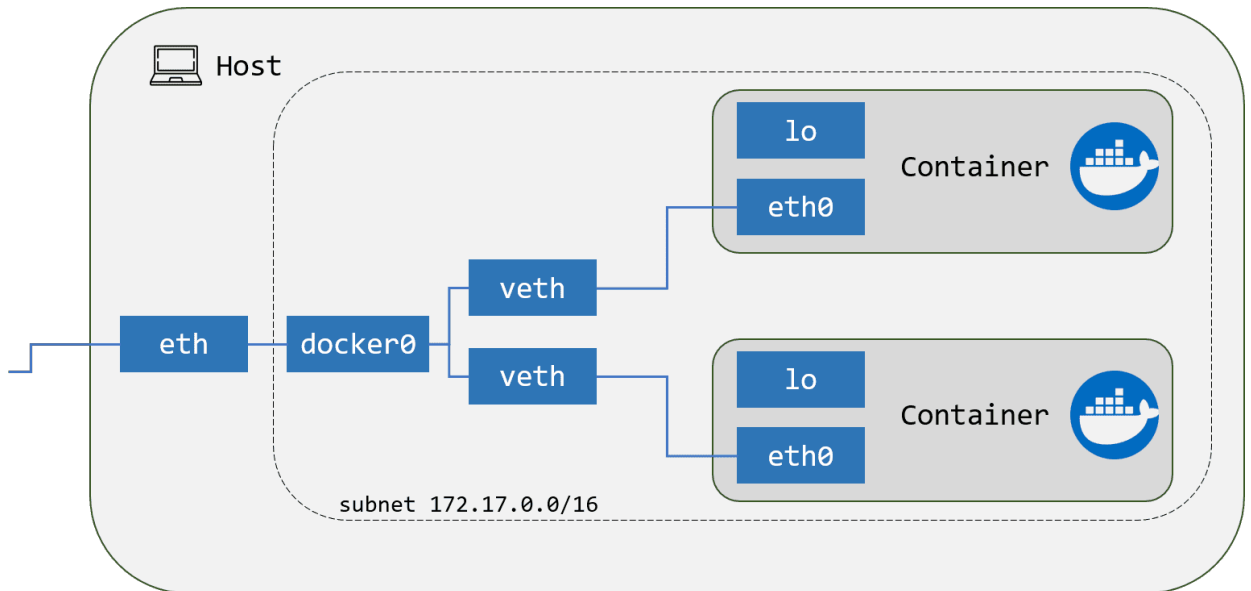
- 인터넷 상의 다른 네트워크와 직접적으로 통신할 수 있습니다.
- 예를 들어, 서울시 영등포구 여의도동 192-1 같이 모두가 인식 가능한 인터넷 상의 주소를 의미합니다.

2. 사설 IP (Private IP)

- 인터넷 상의 다른 네트워크와 직접적으로 통신할 수 없습니다.
- 동일 네트워크에서만 직접적으로 통신할 수 있습니다.
- 예를 들어, 동생 방, 큰 방 같이 내부에서만 인식가능한 주소를 의미합니다.

3. Docker 네트워크 구성

- Docker 내부에 사설 네트워크가 구성됩니다.
- Host 네트워크에 연결하려면 별도의 작업이 필요합니다.



1. Docker hub

- <https://hub.docker.com/>

2. 설치 및 실행

- `docker pull mariadb`
- `docker run --name mariadb-server -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mariadb`
- `docker run --name mariadb-server -p 3306:3306 -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mariadb`

1. 필수 설치 요소

- NodeJs (<https://nodejs.org/en>)
- VSCode (Prettier plugin)

2. 프로젝트 세팅 (Package.json)

- `git clone` [Repository URL]
- `npm install`

3. 프로젝트 실행

- `npm start`

4. Docker 세팅 (Package.json)

- `npm run build:prod`
- `npm run docker:build`
- `npm run docker:run`

1. 필수 설치 요소

- IntelliJ

2. 프로젝트 세팅

- Gradle build

3. Docker 세팅

- `docker build --build-arg PROFILE=local -t osts-api:latest . -f osts-api/Dockerfile`
- `docker run -it -p 8080:8080 osts-api`