

설치

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
sudo ln -s /usr/local/bin/docker-compose
/usr/bin/docker-compose
```

YAML	JSON
YAML Ain't Markup Language	JavaScript Object Notation
주석 사용 가능	주석 사용 불가
한글 등의 유니코드를 그대로 사용 가능	한글 등의 멀티 바이트 문자는 인코딩 수행
주로 환경 구성 등의 설정 파일 작성 시 사용	주로 API 작성 시 사용
<pre>version: '3.8' services: mydb: image: mariadb:10.4.6</pre>	<pre>{ "version": "3.8", "services": { "mydb": { "image": "mariadb:10.4.6" } } }</pre>

주의 사항

- 야물 코드의 계층 구조는 부모-자식 간의 레벨을 들여쓰기로 엄격하게 구분해야 함
- 주의할 것은 들여쓰기를 탭이 아닌 공백으로 정확히 구분해야 한다는 점

```
    자식-2:
      자식-2-1: 설정값
      자식-2-2: 설정값

# 예러1
자식-1:
  자식-1-1: 설정값
  자식-1-1: 설정값
  자식-2: <-- error
  자식-2-1: 설정값
  자식-2-2: 설정값

# 예러2
자식-1:
  자식-1-1: 설정값
  자식-1-1: 설정값
자식-2:
  자식-2-1: 설정값
  자식-2-2: 설정값 <-- error
```

명령어

- container_name
 - 생략 시 자동으로 부여.
 - 디렉터리_서비스명_n으로 생성
 - docker run --name 옵션과 동일

- ports
 - 서비스 내부 포트와 외부 호스트 포트를 지정하여 바인드
 - 외부 노출 포트 지정
 - `docker run -p` 옵션과 동일
- expose
 - 호스트 운영체제와 직접 연결하는 포트를 구성하지 않고 서비스만 포트를 노출
 - 필요시 링크로 연결된 서비스와 서비스 간의 통신만 사용
- networks
 - 최상위 레벨의 networks에 정의된 네트워크 이름을 작성
 - `docker run`의 `--net` 옵션과 동일
- volumes
 - 서비스 내부 디렉터리와 호스트 디렉터리를 연결하여 데이터 지속성 설정
 - `docker run -v` 옵션과 동일
- environment
 - 서비스 내부 환경 변수 설정
 - 환경 변수가 많은 경우에는 파일 `.env` 로 만들어 `env_file` 옵션에 파일명을 지정
 - `docker run -e` 옵션과 동일

- command
 - 서비스가 구동 이후 실행할 명령어 작성
 - docker run의 마지막에 작성되는 명령어
- restart
 - 서비스 재시작 옵션 지정
 - no : 수동 재시작
 - always : 컨테이너 수동 제어를 제외하고 항상 재시작
 - on-failure : 오류가 있을 시 재시작
 - docker run --restart 옵션과 동일
- depends_on
 - 서비스 간의 종속성을 의미하며 먼저 실행해야 하는 서비스를 지정하여 순서 지정
 - 이 옵션에 지정된 서비스가 먼저 시작됨