

도커 이미지 최적화하기: 보안, 용량, 속도

최 혁

이미지 최적화하기

```
# 이미지와 컨테이너, 볼륨, 빌드 캐시 등이 점유하는 실제 디스크 용량  
docker system df
```

꼭 필요한 파일만 이미지에 포함시키기

- 그 뒤에 오는 instruction에서 불필요한 파일을 지웠다 하더라도 이전 레이어에 그대로 남아 디스크 용량을 차지한다.
- .dockerignore 파일에 불필요한 디렉터리나 파일 목록을 기재하면 빌드 컨텍스트에서 이들 파일을 제외할 수 있다.

좋은 기반 이미지 고르는 법

- 운영체제 기반 이미지의 크기가 크면 다양한 도구가 포함되지만, 보안상의 허점이 되기 쉽다.
- OpenJDK 공식 이미지도 설정값에 따라 다양한 태그가 부여된 JRE와 JDK 이미지가 있다.
- 이미지에 방치한 불필요한 파일이 공격 수단으로 될 수 있다.
- 골든 이미지를 사용하여 이러한 문제를 피할 수 있다.
- 앤코어 같은 오픈 소스 도커 이미지 분석 도구를 골든 이미지에 삽입해 빌드 중에 보안 검사를 할 수 있다.
- 앤코어는 이미지에 사용된 라이선스부터 운영체제와 애플리케이션 플랫폼에 대한 상세 정보, 이미지에 포함된 바이너리 파일의 보안 문제까지 알 수 있다.

이미지 레이어 수와 크기 줄이기

불필요한 요소나 설치 후 전재를 통제해야 최소한의 크기와 보안석을 갖출 수 있다.

- 패키지 설치 명령에 추천 패키지를 설치하지 않는 것만으로 이미지 크기를 많이 줄일 수 있다.
- 여러 instruction을 하나로 합쳐 이미지 레이어의 수를 줄일 수 있다.
- 멀티 스테이지 빌드를 활용하여 이미지 크기를 줄일 수 있다.

정리

- 기반 이미지를 잘 고르기(골든 이미지)
- 아주 간단한 애플리케이션이 아니라면 멀티 스테이지 빌드 적용하기
- 불필요한 패키지나 파일을 포함시키지 말고, 레이어 크기를 최소한으로 유지하기
- Dockerfile 스크립트의 instruction은 자주 수정하는 순서대로 뒤에 오도록 배치해 캐시를 최대한 활용하기