

* Container 테스트 방법

만든 날짜: 2020-09-25 오전 5:53

수정한 날짜: 2020-09-25 오전 6:05

컨테이너 외부에서 테스트를 실행

CI Build Test 가 Locally 수행되고 배포후 API 테스트를 실행

- 장점
 - 테스트에 부담이 없음
 - 노가다만 하면 됨
- 단점
 - 테스트 도구를 설정하고 구성 필요
 - Docker Run 환경과 Test 환경이 일치한다고 볼 수 없음

컨테이너 내부에서 테스트를 실행

테스트 환경을 컨테이너에서 사용 가능하도록 함

- 컨테이너에 포함되는 것
 - 필수 패키지
 - 테스트 도구 (특정 버전)
 - 테스트 도구 플러그인
 - 테스트 스크립트
 - 모든 필수 패키지가 포함
- 장점
 - 테스트 도구를 설정하고 구성 할 필요가 없음
 - 동일한 테스트 도구를 사용하여 정확히 동일한 테스트를 실행
- 단점
 - 이미지 크기 증가
 - 테스트 특정 구성으로 이미지 런타임 환경을 오염시키고 불필요한 종속성을 도입
 - 테스트 결과 및 로그로 수행 할 작업을 결정

컨테이너 내부에서 테스트를 실행 - Test / Ops 환경을 구분 가능

Docker 에서 ONBUILD 인스트럭션 사용

1. 원본 Dockerfile에서 임시 Dockerfile.test 생성

- Dockerfile.test 에 포함되어야 하는 것들
 - 테스트 도구 — Phantom.js, Selenium, Chakram, Gatling,...
 - 테스트 도구 런타임 — Node.js, JVM, Python, Ruby,...
 - 테스트 환경 구성 — 환경 변수, 구성 파일, 부트 스트랩 스크립트,...
 - 테스트 — 컴파일 된 패키지 또는 스크립트 파일
 - 테스트 데이터 — 테스트 에 사용되는 모든 종류의 데이터 파일 : json, csv, txt, xml,...
 - 테스트 시작 스크립트 — 테스트를 실행하기위한 시작 스크립트 . 여기에서 테스트 도구를 시작

2. docker build --tag TAG-test -f Dockerfile.test ...

3. docker run -v ./tests/results:/var/tests/results [OPTIONS] IMAGE:TAG-test [COMMAND] [ARG...]