

어디서든 실행할 수 있는 도커 이미지 만들기: 리눅스, 윈도우, 인텔, ARM

최 혁

다중 아키텍처 이미지

- 다중 아키텍처 이미지는 여러 개의 변종 이미지 형태로 빌드되고 레지스트리에 푸시된다.
- 다중 아키텍처 이미지를 내려받으려 시도하면 도커는 현재 컴퓨터의 아키텍처와 운영체제에 일치하는 변종 이미지를 받아온다.
- 이미지를 내려받을 때 도커 엔진이 실행중인 컴퓨터의 운영체제와 아키텍처를 알아낸 후 레지스트리에 이들 정보가 일치하는 이미지 변종이 있는지 확인한다. (일치하는 변종이 없다면 오류가 발생한다)

다중 아키텍처 이미지를 만들기 위한 Dockerfile 스크립트

다중 아키텍처 이미지를 만들 수 있는 방법은 크게 두 가지가 있다.

1. 멀티 스테이지 Dockerfile 스크립트를 이용해 컨테이너에서 소스 코드를 빌드하고 패키징

- 첫 번째 방법의 장점은 Dockerfile 스크립트 하나로 다른 아키텍처의 컴퓨터에서 이미지를 빌드하면 해당 아키텍처의 이미지를 만들 수 있다. (대신 모든 대상 운영체제에서 공통으로 사용할 수 있는 명령어만 사용해야 함)
- 다만 빌드에 사용된 이미지가 다중 아키텍처 이미지가 아니거나 원하는 아키텍처를 모두 지원하지 않는다면 이 방법을 사용할 수 없다.

2. Dockerfile 스크립트를 아키텍처나 운영체제별로 따로 작성

- 만약 첫 번째 방법이 불가능하다면 이 방법을 사용해야 한다.
- Dockerfile 스크립트를 따로 작성하면 Dockerfile 스크립트 instruction이 전혀 달라도 동작은 같도록 할 수 있는 장점이 있다.

다중 아키텍처 이미지를 레지스트리에 푸시하기

- 다중 아키텍처 이미지를 푸시하기 위해선 매니페스트와 함께 이미지를 레지스트리로 푸시해야 한다.
- 매니페스트란 여러 개의 변종 이미지를 하나의 이미지 태그로 묶는 메타데이터를 말한다.
- 매니페스트는 도커 명령행 도구로 만들어 레지스트리에 푸시한다.

이미지에 도커 계정 이름이 포함된 태그를 부여한다.

```
docker image tag diamol/ch16-folder-list:linux-amd64 "$dockerId/  
ch16-folder-list:linux-amd64"
```

```
docker image tag diamol/ch16-folder-list:linux-arm64 "$dockerId/  
ch16-folder-list:linux-arm64"
```

이미지를 도커 허브에 푸시한다(해당 태그를 가진 모든 이미지가 푸시된다)

```
docker image push "$dockerId/ch16-folder-list"
```

생성된 매니페스트 확인

```
docker manifest inspect diamol/base
```

매니페스트 이름과 해당 매니페스트에 포함시킬 모든 이미지 태그를 열거해 매니페스트를 생성한다

```
docker manifest create "$dockerID/ch16-folder-list" ...
```

생성한 매니페스트를 도커 허브에 푸시

```
docker manifest push "$dockerID/ch16-folder-list"
```

도커 Buildx를 사용해 다중 아키텍처 이미지 빌드하기

- Buildx는 docker build 명령의 호가장판으로 최적화된 빌드 엔진이 적용돼 빌드 성능이 뛰어나다.
- 크로스 플랫폼 빌드를 지원하며 도커 컨텍스트와 통합돼 있기에 한 번의 명령으로 여러 대의 서버에서 빌드할 수 있다.
- Buildx는 도커 명령행 도구의 플러그인이기에 사용하려면 plugin 디렉터리에 플러그인 파일을 내려받아 복사해야 한다.
- ```
docker buildx build -t "$dockerId/ch16-folder-list-2" --platform linux/amd64,linux/386 --push
```

## 참고

- Dockerfile 스크립트에 다음 두 가지 사항만 미리 적용해 놓는다면, 큰 어려움 없이 다중 아키텍처 이미지로 전환할 수 있다.
  1. FROM instruction에 항상 다중 아키텍처 이미지를 기반으로 지정
  2. RUN, CMD instruction에는 특정 운영체제에서만 사용되는 명령어를 사용하지 않는 것