

컨테이너의 애플리케이션 설정 관리

최 혁

다단 애플리케이션 설정

설정 데이터의 종류

- 버전에 따라 달라지는 설정(ex 릴리즈 단계 정보)
- 환경에 따라 달라지는 설정(ex 환경 이름)
- 기능 설정(ex 통계 수집 여부)

저자가 제시하는 패턴: 이미지에 포함된 기본 설정 파일(버전 정보), 볼륨으로 마운트된 로컬 디렉터리의 오버라이드 설정 파일(환경 설정), 환경변수(기능 설정)를 병합하는 패턴

환경별 설정 패키징하기

여러 가지 애플리케이션 프레임워크에서 환경별 설정 파일을 모두 배포에 포함시킬 수 있는 기능을 제공한다.

- application.yaml
- application-{환경}.yaml
- 환경 변수(환경 이름 정의)

저자는 이미지에 설정 파일을 모두 포함시키는 방법은 서버 이름, URL, 파일 경로, 로그 수준 등의 정보가 노출되면 보안에 취약해지는 것을 우려하여 선호하지 않는다

런타임에서 설정 읽어들이기

Go 언어는 Viper라는 설정 모듈이 널리 쓰인다.

1. 이미지에 포함된 config 디렉터리에서 기본 설정 파일을 읽어들인다.
2. 환경별 파일은 config-override에서 읽어들이는데, 이 디렉터리는 이미지에서 빈 디렉터리이며 파일 시스템 마운트로 외부에 주입된다.
3. 환경 변수는 설정 파일의 설정값을 오버라이드할 수 있다.
 - 바이퍼의 설정 파일 포맷은 JSON 또는 YAML 또는 TOML이 널리 쓰인다.
 - TOML은 클라우드 환경에 장점이 크다.

레거시 애플리케이션에 설정 전략 적용하기

Dockerfile 스크립트로 설정하기

1. 컨테이너에 지정된 오버라이드 설정 파일 읽어들이기
2. 환경 변수에서 오버라이드 설정을 읽어들이기
3. 오버라이드 설정 파일과 환경 변수를 병합하기(환경 변수값 우선)
4. 병합된 오버라이드 설정을 컨테이너 내 대상 설정 파일에 추가하기

```
docker run -d -p 8091:80 -v "$(pwd)/config/dev:/config-override" -e  
    CONFIG_SOURCE_PATH="/config-override/application.properties" -e  
    IOTD_ENVIRONMENT="custom" diamol/ch18-image-of-the-day
```