

출처: <https://qiita.com/t-kuni/items/71bc73429643ef727011>

Windows + Docker 개발 환경에서 응용 프로그램이 무거운 문제를 unison라는 도구로 해결했기 때문에 공유한다.

인터넷에 일부 unison을 사용하고 있는 문서가 있는데 더 쉽게 도입 할 수 있고, 응용 프로그램에 영향이 적은 방법을 알게 되어 기록을 남긴다.

덧붙여서 나는 Laravel 응용 프로그램 개발에서 이 환경을 사용하고 있다.

3개월 정도 운영하고 있으며, 일반적으로 개발하고데 큰 문제가 없다.

composer에서 패키지를 모두 업그레이드 했을 때 동기화가 이상해진 정도이다.

일단 장단점은 아래와 같다

장점

응용 프로그램 실행이 빨라진다

단점

작업 시작 시 unison 클라이언트를 시작하는 사람 수고가 증가

또한, docker-compose를 사용한다는 전제로 구성되어 있다.

대략의 구성에 대해

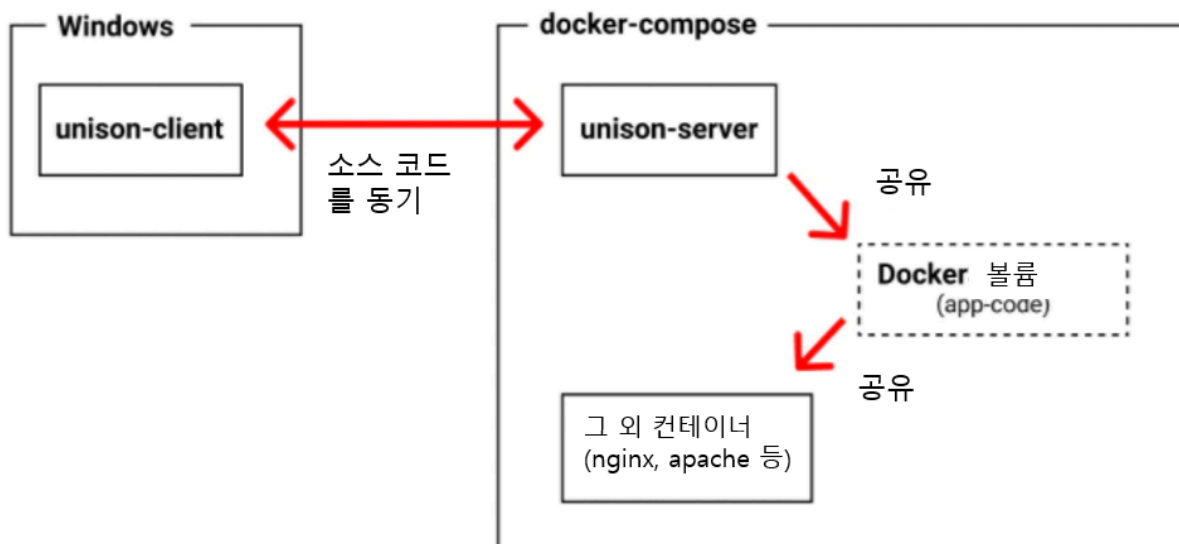
[unison](#) 이라는 파일 동기화 도구를 사용한다

아래 그림과 같이 unison 클라이언트를 Windows쪽에 설치하고, unison 서버를 docker-compose 서비스 중 하나로 시작한다.

이 두 unison 간의 소스 코드를 동기화 한다.

(기존의 docker-compose.yml에 unison 서버를 추가 할 수 있다.)

unison에서 동기화된 소스 코드를 Docker 볼륨을 통해 다른 컨테이너에 공유한다.



샘플 코드를 준비했다

본 기사의 구성을 즉시 시험 할 수 있는 샘플을 준비했다.

자세한 내용은 저장소의 README를 참조한다.

저장소: <https://github.com/t-kuni/docker-unison-example>

Unison 서버를 시작한다

docker-compose.yml에 unison 서버를 추가한다.

```
version: '3'
services:
  # Unison 서버
  unison:
    image: onnimonni/unison
    environment:
      - UNISON_DIR=/tmp/unison # 동기화된 파일을 쓰는 폴더를 지정한다
    ports:
      - "5000:5000"
    volumes:
      - app-code:/tmp/unison # Docker 볼륨을 마운트하는 폴더를 지정한다
    restart: always
  # 그 외 컨테이너
  apache:
    image: httpd:2.4
    volumes:
      - app-code:/usr/local/apache2/htdocs/ # 그 외 컨테이너에도 Docker
      볼륨 경유로 소스코드를 공유한다
    ports:
      - "8080:80"
volumes:
  app-code:
```

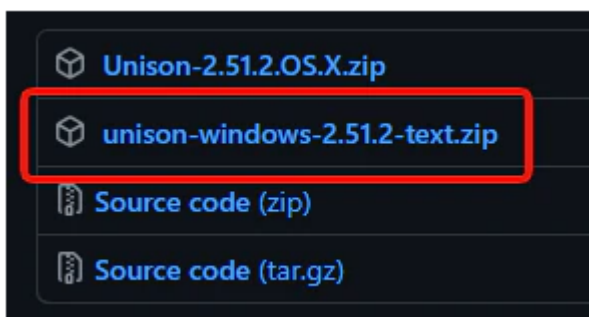
준비가되면 `docker-compose up -d` 로 컨테이너를 시작한다

Unison 클라이언트를 다운로드한다

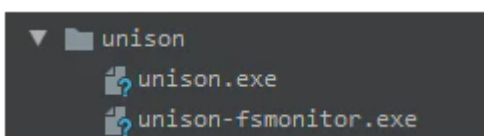
아래 페이지에서 unison 클라이언트를 다운로드한다.

unison 클라이언트 : <https://github.com/bcpierce00/unison/releases>

서버의 버전과 일치해야하기 때문에 이번에는 2.51.2 클라이언트를 다운로드한다.



docker-compose.yml 와 동일한 계층에 unison 폴더를 생성하고 unison 클라이언트를 배치한다.



Unison 클라이언트를 시작한다

unison 클라이언트를 시작하는 bat 파일을 준비한다.

docker-compose.yml 와 같은 폴더에 sync.bat 라는 이름으로 아래의 코드를 작성한다.

```
:LOOP
    .\unison\unison . socket://192.168.99.100:5000/ ^
        -repeat watch ^
        -auto ^
        -batch ^
        -prefer newer ^
        -ignore "Path .git" ^
        -ignore "Path .idea" ^
        -ignore "Path node_modules"
goto :LOOP
```

위의 코드는 unison을 시작하여 현재 폴더를 동기화 하고 있다.

IP 주소 부분(192.168.99.100)는 컨테이너의 IP 주소를 입력한다.

또한 드물게 클라이언트가 떨어지는 경우가 있기 때문에 무한 루프로 다시 시작하게 한다

unison 옵션에 대해

unison 옵션의 의미는 아래와 같다.

-repeat watch

파일을 모니터링하고 변경이 있으면 동기화한다

-auto

충돌 시의 확인을 자동으로 허용

-batch

런타임 시 확인을 하지 않는다

prefer newer

충돌 시 해결 방법으로 '더 새로운 것'을 우선한다

ignore

동기화 하지 않는 파일을 지정한다.

ignore 옵션은 필요에 따라 추가한다.

예를 들어 Laravel 환경이라면 storage 아래의 일부 파일은 제외하는 것이 좋다.

다른 옵션은 [공식 문서](#)를 참조한다.

준비가되면 sync.bat를 시작한다.

나머지는 자동으로 동기화를 해준다.

동작 확인

아래 코드를 index.html로 docker-compose.yml와 같은 폴더에 배치하고, unison으로 동기화한다.

```
<! DOCTYPE html>
<html lang = "ja" >
<head>
  <meta charset = "utf-8" >
  <title> title </ title>
</ head>
<body>
  <h1> Hello unison! </ h1 >
  <p> 이 메시지가 표시되면 unison에서의 동기화에 성공하고 있습니다 . </ p>
  <p> index.html을 다시 재 동기화되는지 시도해보십시오 . </ p>
</ body>
</ html>
```

192.168.99.100:8080(IP 주소는 컨테이너의 IP로 대체)에 연결하고 아래 화면이 표시되면 OK 이다.

동기화가 깨졌을 때는?

동기화에 사용하는 Docker 볼륨(샘플에서는 app-code)을 docker volume rm 으로 삭제하고, 다시 동기화한다.