



## 인셉션

*요약: 이 문서는 시스템 관리와 관련된 연습 문제입니다.*

*버전: 2.1*

# 콘텐츠

I	서문	2
II	소개	3
III	일반 가이드라인	4
IV	필수 부분	5
V	보너스 부분	9
VI	제출 및 동료 평가	10



## 제1장 서문



## 2장 소개

이 프로젝트는 Docker를 사용하여 시스템 관리에 대한 지식을 넓히는 것을 목표로 합니다. 여러 개의 Docker 이미지를 가상화하여 새로운 개인 가상 머신에 생성합니다.

## 제3장 일반 가이드

### 라인

- 이 프로젝트는 가상 머신에서 수행해야 합니다.
- 프로젝트 구성에 필요한 모든 파일은 `srcs`에 배치해야 합니다.  
폴더로 이동합니다.
- 메이크파일도 필요하며 디렉터리의 루트에 위치해야 합니다. 이 파일은 전체 애플리케이션을 설정해야 합니다(즉, `docker-compose.yml`을 사용하여 Docker 이미지를 빌드해야 합니다).
- 이 주제는 여러분의 배경 지식에 따라 아직 배우지 않았을 수도 있는 개념을 실제로 적용해야 합니다. 따라서 이 과제를 완료하는 데 도움이 될 만한 Docker 사용법과 관련된 많은 문서와 기타 유용한 정보를 주저하지 말고 읽어보시기 바랍니다.



## 4장 필수 부분

이 프로젝트는 특정 규칙에 따라 다양한 서비스로 구성된 소규모 인프라를 설정하는 것으로 구성됩니다. 전체 프로젝트는 가상 머신에서 수행해야 합니다. 도커 컴포즈를 사용해야 합니다.

각 Docker 이미지는 해당 서비스와 동일한 이름을 가져야 합니다.

각 서비스는 전용 컨테이너에서 실행해야 합니다.

성능 문제를 위해 컨테이너는 두 번째로 안정적인 버전의 Alpine 또는 Debian으로 빌드해야 합니다. 선택은 여러분의 몫입니다.

또한 서비스당 하나씩 자체 Docker파일을 작성해야 합니다. Docker파일은 Makefile에서 docker-compose.yml을 호출해야 합니다.

즉, 프로젝트의 Docker 이미지를 직접 빌드해야 합니다. 그런 다음 기성 Docker 이미지를 가져오는 것은 물론 DockerHub와 같은 서비스를 사용하는 것도 금지됩니다 (Alpine/Debian은 이 규칙에서 제외됨).

그런 다음 설정해야 합니다:

- TLSv1.2 또는 TLSv1.3만 있는 NGINX가 포함된 Docker 컨테이너입니다.
- 워드프레스 + php-fpm(설치 및 구성해야 함)이 포함된 도커 컨테이너로, nginx 없이 워드프레스만 포함되어야 합니다.
- nginx 없이 MariaDB만 포함된 Docker 컨테이너입니다.
- 워드프레스 데이터베이스가 들어 있는 볼륨입니다.
- 워드프레스 웹사이트 파일이 들어 있는 두 번째 볼륨입니다.
- 컨테이너 간의 연결을 설정하는 도커 네트워크입니다.



충돌이 발생하면 컨테이너를 다시 시작해야 합니다.



Docker 컨테이너는 가상 머신이 아닙니다. 따라서 이를 실행하려고 할 때 'tail -f' 등을 기반으로 하는 해킹 패치를 사용하지 않는 것이 좋습니다. 데몬의 작동 방식과 데몬을 사용하는 것이 좋은지 아닌지에 대해 읽어보세요.



물론 네트워크: 호스트 또는 --link 또는 links: 사용은 금지되어 있습니다. 네트워크 줄은 docker-compose.yml 파일에 있어야 합니다. 컨테이너는 무한 루프를 실행하는 명령으로 시작해서는 안 됩니다. 따라서 이는 엔트리포인트로 사용되거나 엔트리포인트 스크립트에 사용되는 모든 명령에도 적용됩니다. 다음은 몇 가지 금지된 해킹 패치입니다: tail -f, bash, sleep infinity, true 동안.

## 인셉션



PID 1과 Docker파일 작성 모범 사례에 대해 읽어보세요.

- 워드프레스 데이터베이스에는 두 명의 사용자가 있어야 하며, 그 중 한 명은 광고 관리자여야 합니다. 관리자의 사용자 이름에는 관리자/관리자 또는 관리자-관리자/관리자를 포함할 수 없습니다(예: 관리자, 관리자, 관리자, 관리자-123 등).



볼륨은 Docker를 사용하는 호스트 머신의 /home/login/data 폴더에서 사용할 수 있습니다. 물론 로그인을 사용자 계정으로 바꿔야 합니다.

작업을 더 간단하게 하려면 도메인 네임이 로컬 IP 주소를 가리키도록 구성해야 합니다. 이 도메인 이름은 login.42.fr이어야 합니다. 다시 말하지만, 자신의 로그인을 사용해야 합니다.

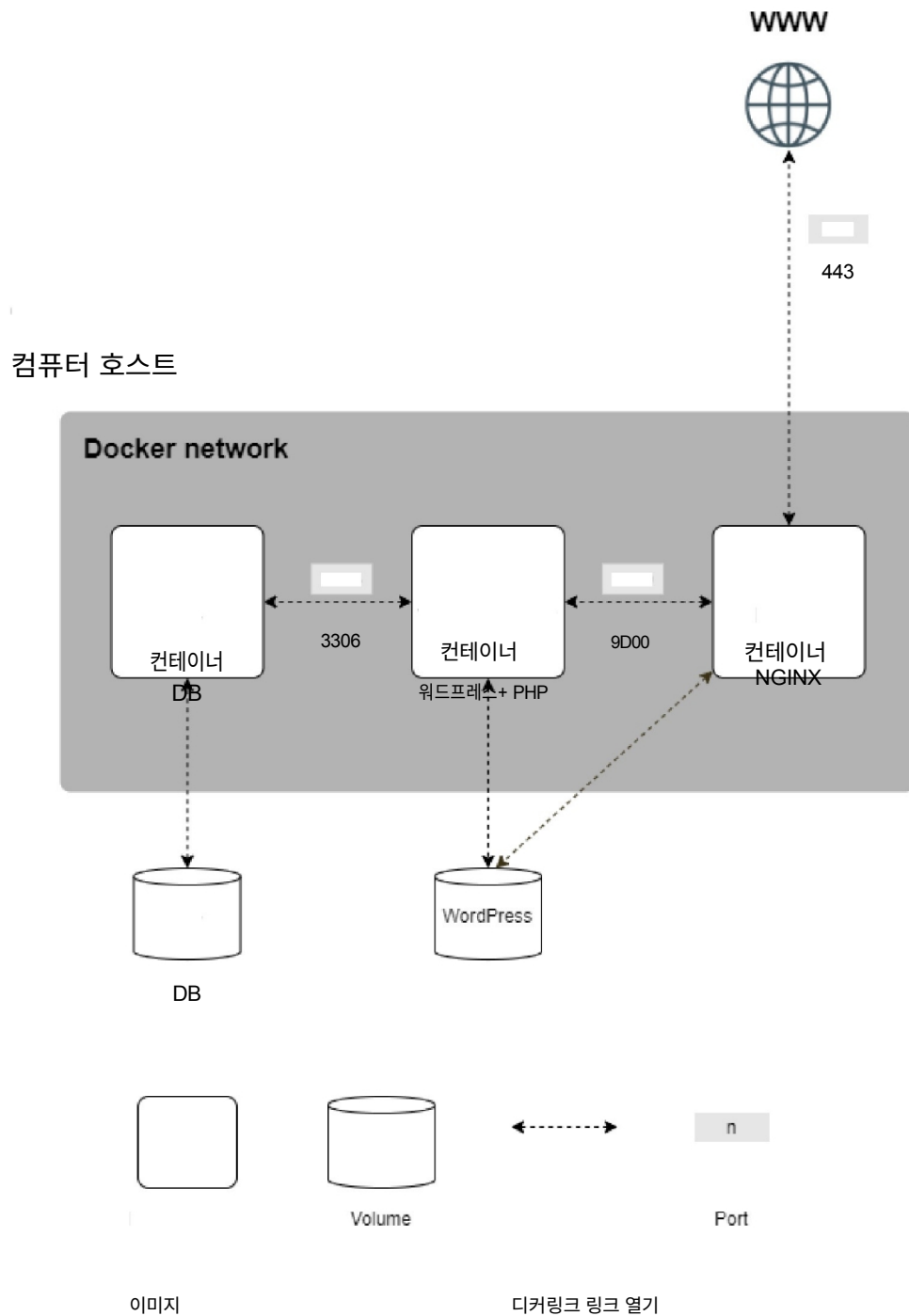
예를 들어 로그인 아이디가 wil인 경우 wil.42.fr은 wil의 웹사이트를 가리키는 IP 주소로 리디렉션됩니다.



최신 태그는 금지되어 있습니다.  
Docker파일에 비밀번호가 없어야 합니다. 환경 변수를 반드시 사용해야 합니다.  
또한 환경 변수를 저장할 때는 .env 파일을 사용하는 것이 좋습니다. .env 파일은 srcs 디렉터리의 루트에 위치해야 합니다.  
NGINX 컨테이너는 포트 443을 통해서만 인프라로 들어가는 유일한 진입점이어야 하며, TLSv1.2 또는 TLSv1.3 프로토콜을 사용해야 합니다.



다음은 예상 결과의 예시 다이어그램입니다:





## 인셉션

다음은 예상되는 디렉토리 구조의 예입니다:

```
$> ls -alR 합
계 XX
DRWXRWXR-X 3 WIL WIL 4096 AVRIL 42 20:42 .
DRWXRWXRWT 17 WIL WIL 4096 AVRIL 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 메이크파일
DRWXRWXR-X 3 WIL WIL 4096 AVRIL 42 20:42 SRCS

./srcs:
총 XX
DRWXRWXR-X 3 WIL WIL 4096 AVRIL 42 20:42 .
DRWXRWXR-X 3 WIL WIL 4096 AVRIL 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 docker-compose.yml
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .env
DRWXRWXR-X 5 WIL WIL 4096 AVRIL 42 20:42 요구 사항

./srcs/요구사항:
총 XX
DRWXRWXR-X 5 WIL WIL 4096 에이브 42 20:42 .
DRWXRWXR-X 3 WIL WIL 4096 에이브 42 20:42 ..
DRWXRWXR-X 4 WIL WIL 4096 에이브 42 20:42 보너스
DRWXRWXR-X 4 WIL WIL 4096 에이브 42 20:42 마리아드
DRWXRWXR-X 4 WIL WIL 4096 에이브 42 20:42 nginx
DRWXRWXR-X 4 WIL WIL 4096 에이브 42 20:42 도구
DRWXRWXR-X 4 WIL WIL 4096 AVRIL 42 20:42 워드프레스

./srcs/requirements/mariadb:
총 XX
DRWXRWXR-X 4 WIL WIL 4096 AVRIL 42 20:45 .
DRWXRWXR-X 5 WIL WIL 4096 AVRIL 42 20:42 ...
DRWXRWXR-X 2 WIL WIL 4096 AVRIL 42 20:42 CONF
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 도커파일
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
DRWXRWXR-X 2 WIL WIL 4096 AVRIL 42 20:42 TOOLS
[...]

./srcs/requirements/nginx:
총 XX
DRWXRWXR-X 4 WIL WIL 4096 AVRIL 42 20:42 .
DRWXRWXR-X 5 WIL WIL 4096 AVRIL 42 20:42 ...
DRWXRWXR-X 2 WIL WIL 4096 AVRIL 42 20:42 CONF
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 도커파일
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
DRWXRWXR-X 2 WIL WIL 4096 AVRIL 42 20:42 도구
[...]

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr
# 인증서
CERTS = /XXXXXXXXXX #

MYSQL 설정
MYSQL_ROOT_PASSWORD=XXXXXXXXXXXXX
MYSQL_USER=XXXXXXXXXXXXX
MYSQL_PASSWORD=XXXXXXXXXXXXX [...]
$>
```



명백한 보안상의 이유로 모든 자격증명, API 키, 환경 변수 등은 .env 파일에 로컬로 저장해야 하며 git에서는 무시해야 합니다. 공개적으로 저장된 자격 증명은 프로젝트 실패로 직결됩니다.



## 5장 보너스 부분

이 프로젝트의 경우 보너스 부분은 단순히 만들려고 합니다.

각 추가 서비스에 대해 도커파일을 작성해야 합니다. 따라서 각 서비스는 자체 컨테이너 내에서 실행되며 필요한 경우 전용 볼륨을 갖게 됩니다.

보너스 목록:

- 캐시를 올바르게 관리하려면 워드프레스 웹사이트에 redis 캐시를 설정하세요.
- 워드프레스 웹사이트의 볼륨을 가리키는 FTP 서버 컨테이너를 설정합니다.
- PHP를 제외한 원하는 언어로 간단한 정적 웹사이트를 만듭니다(예, PHP는 제외됩니다!). 예를 들어, 쇼케이스 사이트나 이력서 발표용 사이트를 만들 수 있습니다.
- 관리자를 설정합니다.
- 유용하다고 생각되는 서비스를 설정합니다. 방어하는 동안 자신의 선택을 정당화해야 합니다.



보너스 부분을 완료하기 위해 추가 서비스를 설정할 수 있습니다. 이 경우 필요에 따라 더 많은 포트를 열 수 있습니다.



보너스 부분은 필수 부분이 완벽한 경우에만 평가됩니다. 완벽하다는 것은 필수 부분을 완벽하게 완료하고 오작동 없이 작동한다는 의미입니다. 모든 필수 요건을 통과하지 못한 경우 보너스 부분은 전혀 평가되지 않습니다.





## 제6장

# 제출 및 동료 평가

평소처럼 Git 저장소에 과제를 제출합니다. 방어 기간 동안에는 저장소 내의 작업만 평가됩니다. 폴더와 파일 이름이 정확한지 다시 한 번 확인하시기 바랍니다.



16D85ACC441674FBA2DF65190663EC3C3C258FEA065D090A715F1B62F5A57F0B75403  
61668BD6823E2F873124B7E59B5CE94BB7ABD71CD01F65B959E14A3838E414F1E871  
F7D91730B

