

# [README] Image Provider Build & Deploy Manual

📋 순번	114
🔗 구분	Basic
🌟 개발 상태	완료
🌟 front 연결	완료

## Carebot Project Build & Deploy

### 5. Backend - Image Provider

#### 사용된 기술



분야	사용한 기술
Program Language	Python 3.11.9
Server Architecture	FastAPI 0.115.6
ASGI Web Server	Uvicorn 0.34.0
Database	MariaDB 10.3.23
Cached Point	Nginx 1.18.0
DB Library	SQLAlchemy 2.0.37

#### 사용되는 환경 변수

```
// .env file
DB_HOST=stg-yswa-kr-practice-db-master.mariadb.database.azure.com
DB_PORT=3306
DB_USER=S12P11A102@stg-yswa-kr-practice-db-master.mariadb.database.azure.com
DB_PASSWORD=#####
DB_SCHEMA=S12P11A102
DB_CHARSET=utf8
IS_DEV=1
IS_DEPLOY=1
SESSION_EXPIRE_TIME=1800
EXTENDED_SESSION_EXPIRE_TIME=259200
REMEMBER_EXPIRE_TIME=2592000
SESSION_CLEANUP_INTERVAL=600
EXTERNAL_TIMEOUT=120.0
MAX_IMAGE_SIZE=32
CACHE_DURATION=3600
```

## 빌드 방법



이 프로젝트는 Docker를 이용한 빌드 과정을 수행합니다. **사전에 Docker와 Docker Compose를 설치하는 과정이 필요합니다.**

1. 우선, 해당 Repository를 Clone하여 모든 파일을 받습니다.
2. `/BE/Image-server` 로 이동합니다.
3. `.env` 파일을 생성하고 위의 내용을 추가합니다.
4. 이미 실행 중인 Container가 있으면, 컨테이너를 끄고 이미지를 삭제하기 위해 아래의 명령어를 입력합니다. 없다면 건너 뛰어도 됩니다.

```
docker compose -f "docker-compose-dev.yml" down --rmi all
```

5. 이후에 Docker Image를 Build하고, Container를 실행하기 위해 다음의 명령어를 입력합니다.

```
docker compose -f "docker-compose-dev.yml" up -d --build
```

6. Container가 생성되었다는 아래의 콘솔 로그를 확인하고, Host의 **127.0.0.1:4185** 에서 Main Page를 접근할 수 있게 됩니다.

```
Network image-server_default Creating
Network image-server_default Created
Container image-provider Creating
Container image-provider Created
Container image-provider Starting
Container image-provider Started
```

## 유의 사항

1. 환경 변수 정보는 배포 환경에 따라 다음과 같이 변경합니다.

변수 이름	내용
<b>DB_HOST</b>	MariaDB 또는 MySQL Database 서버의 주소
<b>DB_PORT</b>	MariaDB 또는 MySQL Database 서버의 포트 번호
<b>DB_USER</b>	MariaDB 또는 MySQL Database 서버의 사용자 ID
<b>DB_PASSWORD</b>	MariaDB 또는 MySQL Database 서버의 사용자 비밀번호
<b>DB_SCHEMA</b>	MariaDB 또는 MySQL Database 서버의 접근 공간
<b>DB_CHARSET</b>	MariaDB 또는 MySQL Database 서버의 인코딩 방식
<b>IS_DEPLOY</b>	배포용으로 빌드 시 → <b>1</b> , 개발용으로 빌드 시 → <b>0</b>

2. Nginx 설정 시, SSL 인증서를 이용한 **HTTPS로 접근하는 방식**을 추천합니다. (민감한 개인정보가 Body에 담겨서 API에 보내지고 있기 때문입니다)