

# Docker

☰ Tags

현재 소프트웨어 연구실에서 마이크로서비스 관련 주제로 과제를 진행하는 중이다. 저번주에는 Kong Gateway, Spring Cloud Gateway가 무엇인지에 대해 조사했었고, 이번주는 직접 마이크로서비스를 Gateway를 사용해서 구현해보기로 하였다.

이때까지 계속 프론트엔드 개발 위주로 해온 나는 백엔드에 대해 모르는 게 참 많지만 프론트를 잘 하기 위해서 백엔드 지식도 어느정도 갖추 필요성을 느껴 차근차근 공부해보고자 한다.

## Docker 의 개념

Docker는 리눅스 컨테이너를 만들고 사용할 수 있도록하는 컨테이너화 기술이다. Docker를 사용하면 컨테이너를 매우 가벼운 모듈화된 가상머신처럼 다룰 수 있다. 컨테이너를 구축, 배포, 복사하고 여러 환경에서 유연하게 사용할 수 있어 클라우드 환경에 최적화되어있다.

## Docker 의 동작 방식

커널의 기능을 사용해 프로세스를 분리하여 독립적으로 실행가능하게 한다. 여러 프로세스와 애플리케이션을 서로 개별적으로 실행하여 인프라를 효과적으로 활용하고 개별 시스템을 사용할 때와 동일한 보인을 유지할 수 있다.

이미지 기반 배포모델을 제공하여 여러 환경에서 애플리케이션 또는 서비스를 모든 종속적인 것들과 쉽게 공유가 가능하다. 컨테이너 환경 내에 애플리케이션 또는 앱을 구성하는 결합된 프로세스의 배포를 자동화한다.

리눅스 컨테이너 기반으로 구축되어 쉽게 앱에 접근하여 빠른 배포와 버전 관리를 할 수 있다.

# Docker와 기존 리눅스 컨테이너의 차이점

처음엔 Docker 기술은 LXC 기술을 기반으로 구축되었으나 이후에 종속 관계를 벗어났다. LXC는 경량의 가상화 방법으로 유용하게 사용되었으나 사용자들에게 좋은 경험을 제공하지 못했다. Docker 기술은 컨테이너 실행뿐만 아니라 생성 및 구축, 이미지 전송, 이미지 버전 관리 프로세스를 용이하게 한다.

전통적인 리눅스 컨테이너는 여러개의 프로세스를 관리할 수 있는 초기화 시스템을 사용한다. = 전체 애플리케이션을 하나로 실행할 수 있다.

Docker 기술은 애플리케이션이 개별 프로세스로 세분화되도록 하며 이를 수행할 수 있는 툴을 제공한다.

## Docker 컨테이너의 이점

- 모듈성
- 계층 및 이미지 버전 제어
- 롤백
- 신속한 배포

## Docker Mac에 설치하기

### Homebrew로 설치하는 방법

다음과 같은 명령어를 사용해 설치한다.

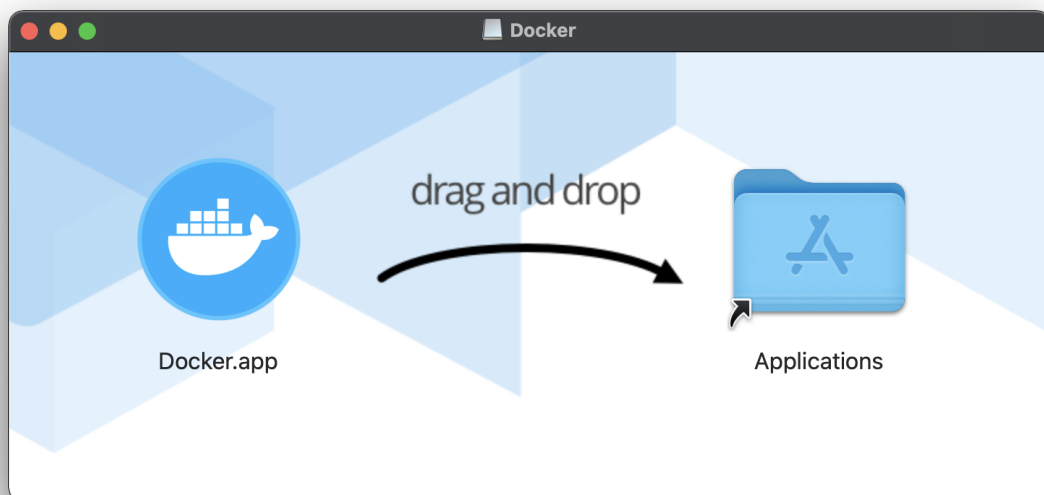
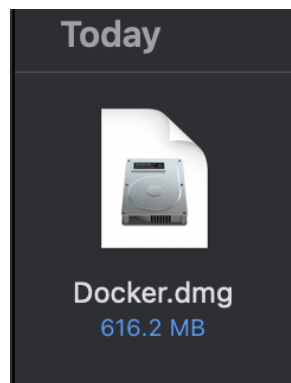
```
brew cask install docker
```

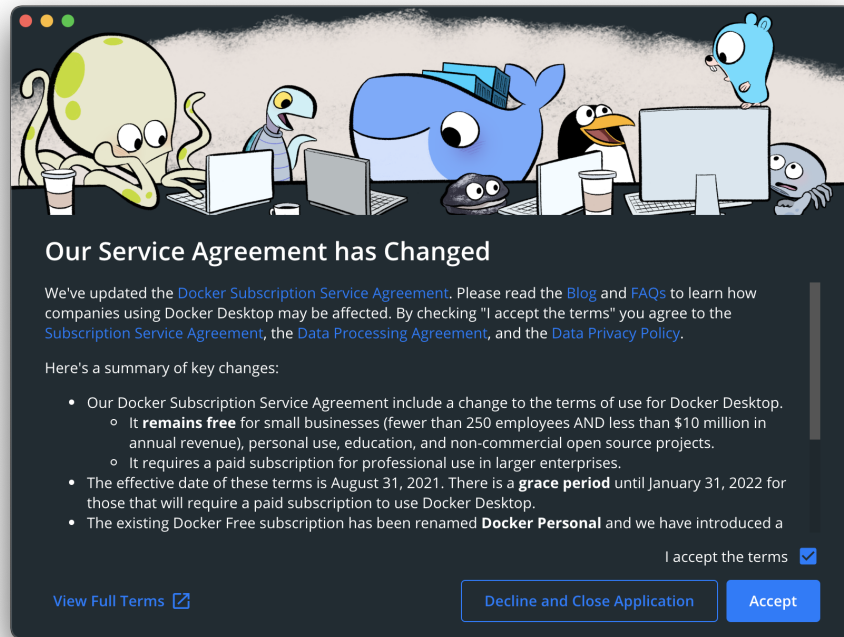
`docker --version` 명령어로 설치가 성공적으로 되었는지 확인한다.

## 설치파일 다운 받아서 설치하는 방법

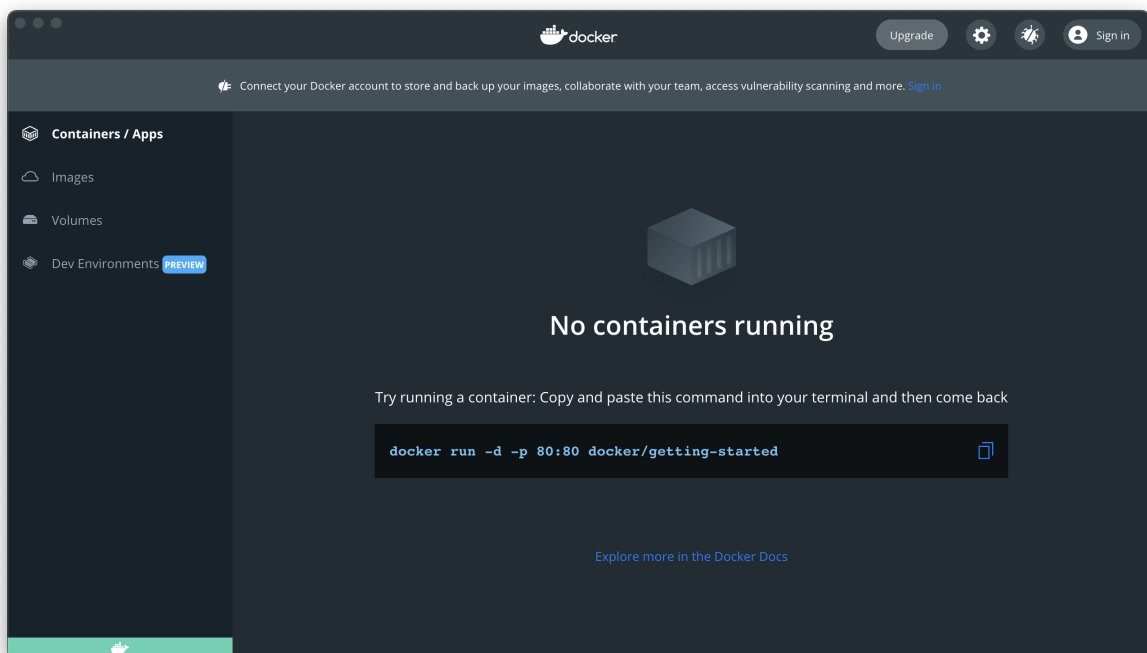
다음 링크에 들어가 설치파일을 다운로드 하고 설치를 진행한다.

<https://docs.docker.com/get-docker/>





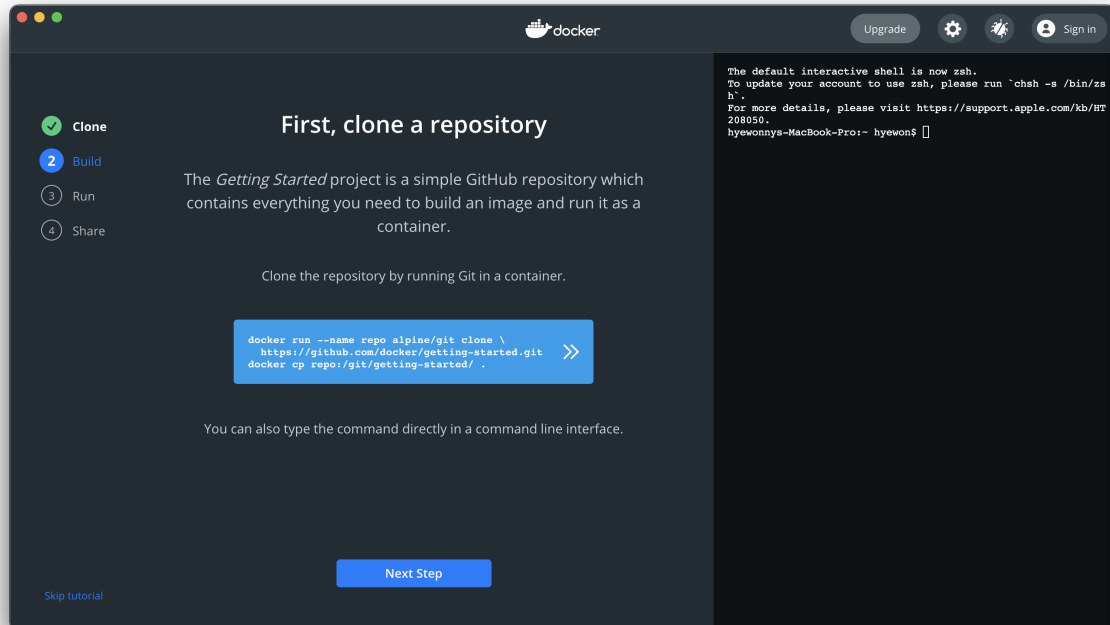
약관에 동의한다.



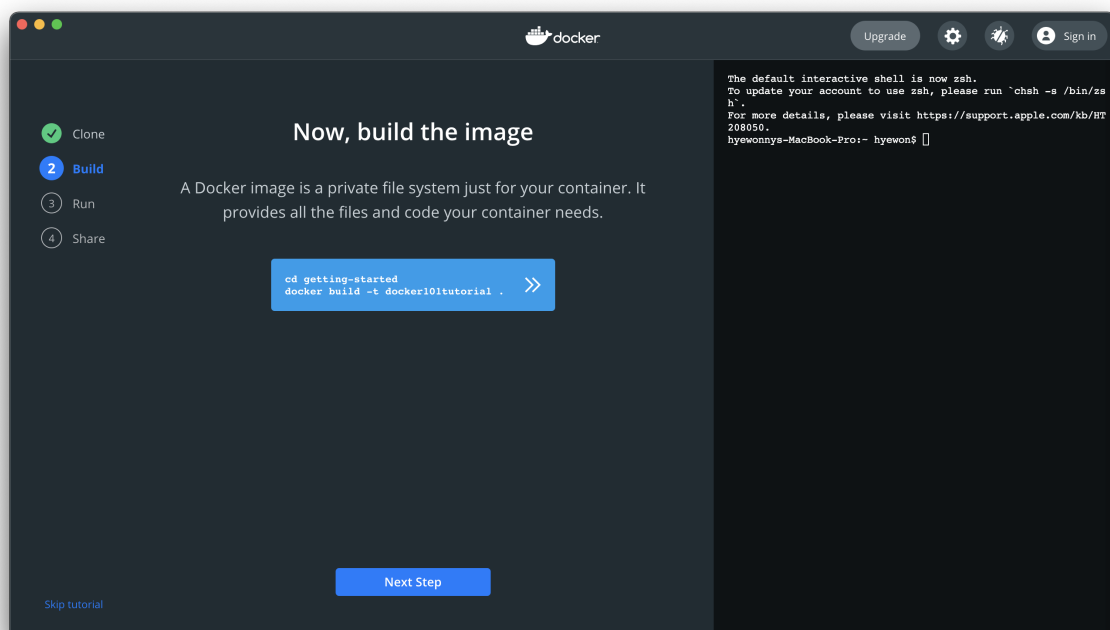
그럼 다음과 같이 창이 뜬다.

# Docker tutorial

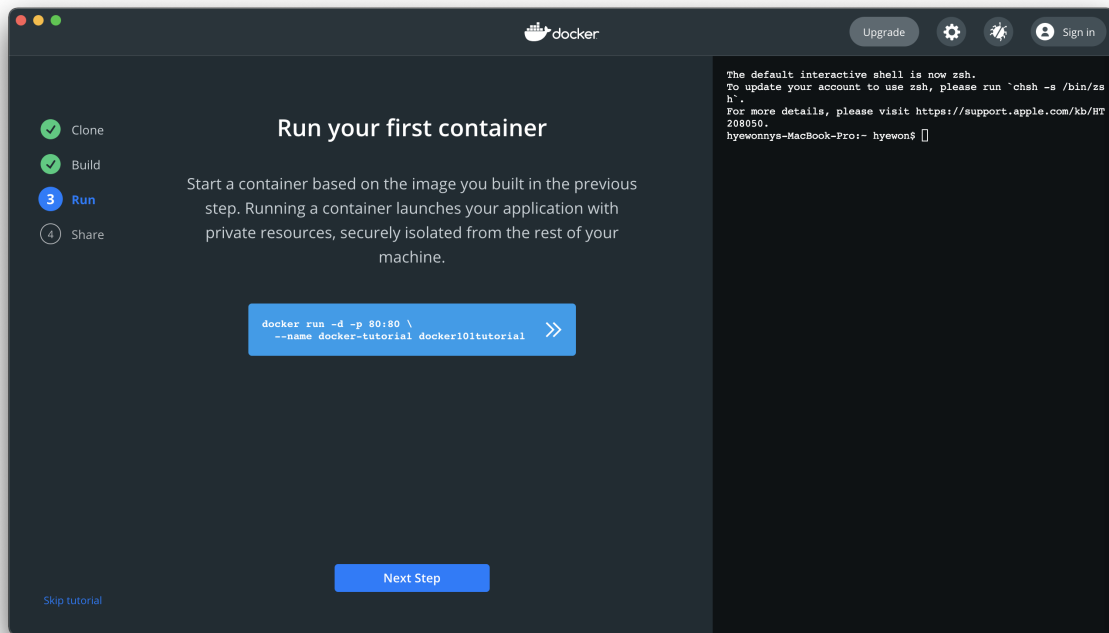
## 1. repository clone



## 2. image 빌드하기



### 3. 첫 컨테이너를 실행하기



### 4. 이미지 저장하고 공유하기

