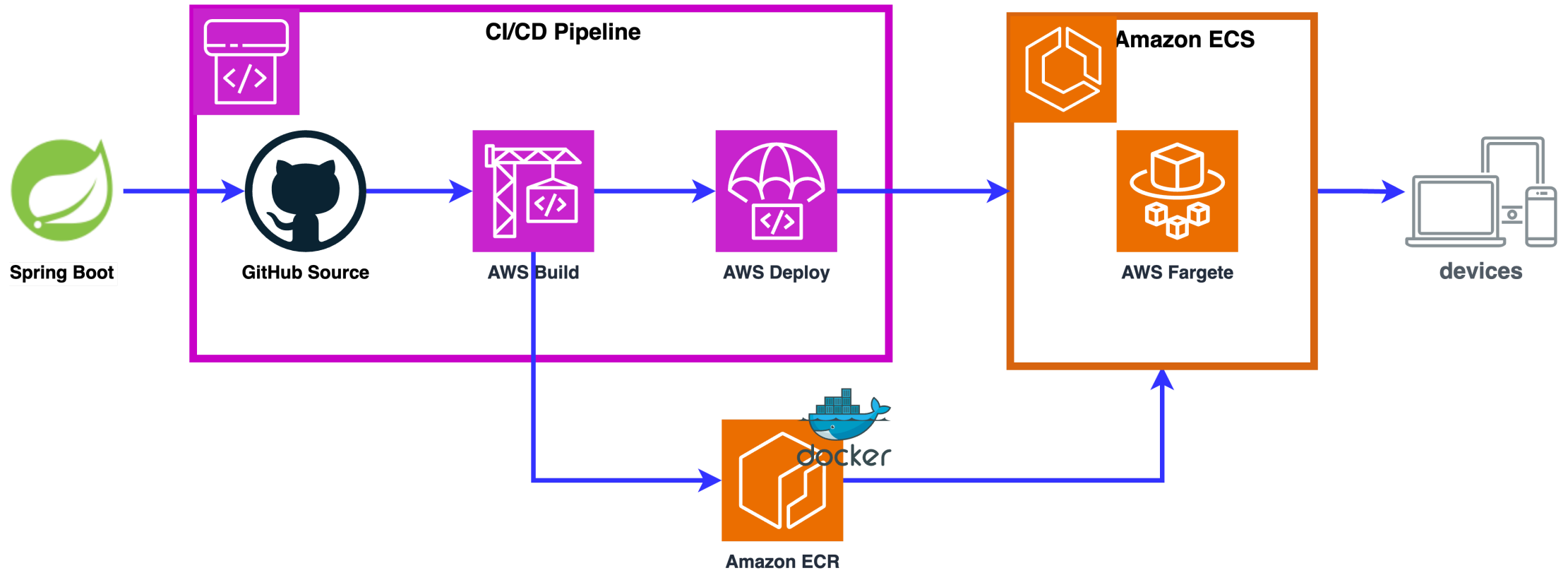


# 전체 아키텍처



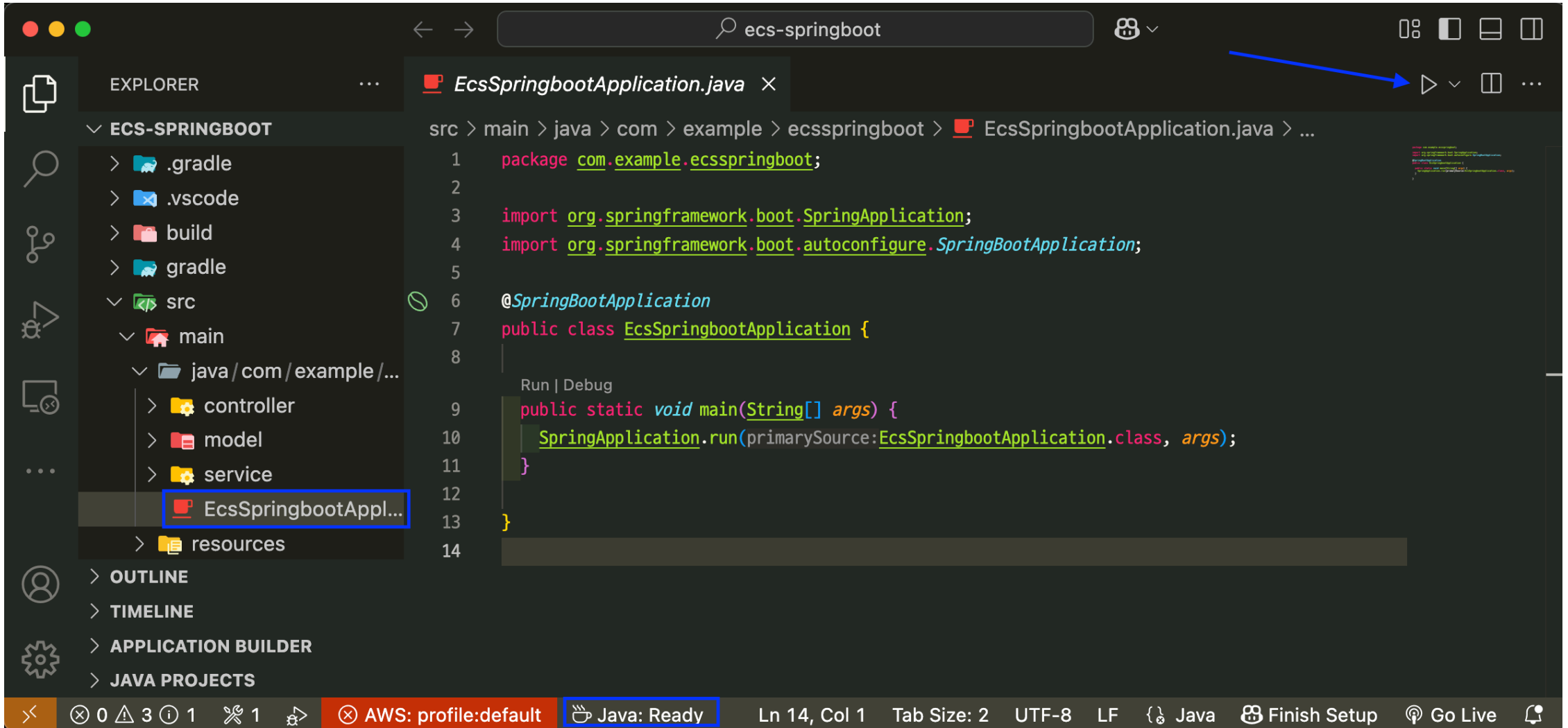
# Spring Boot

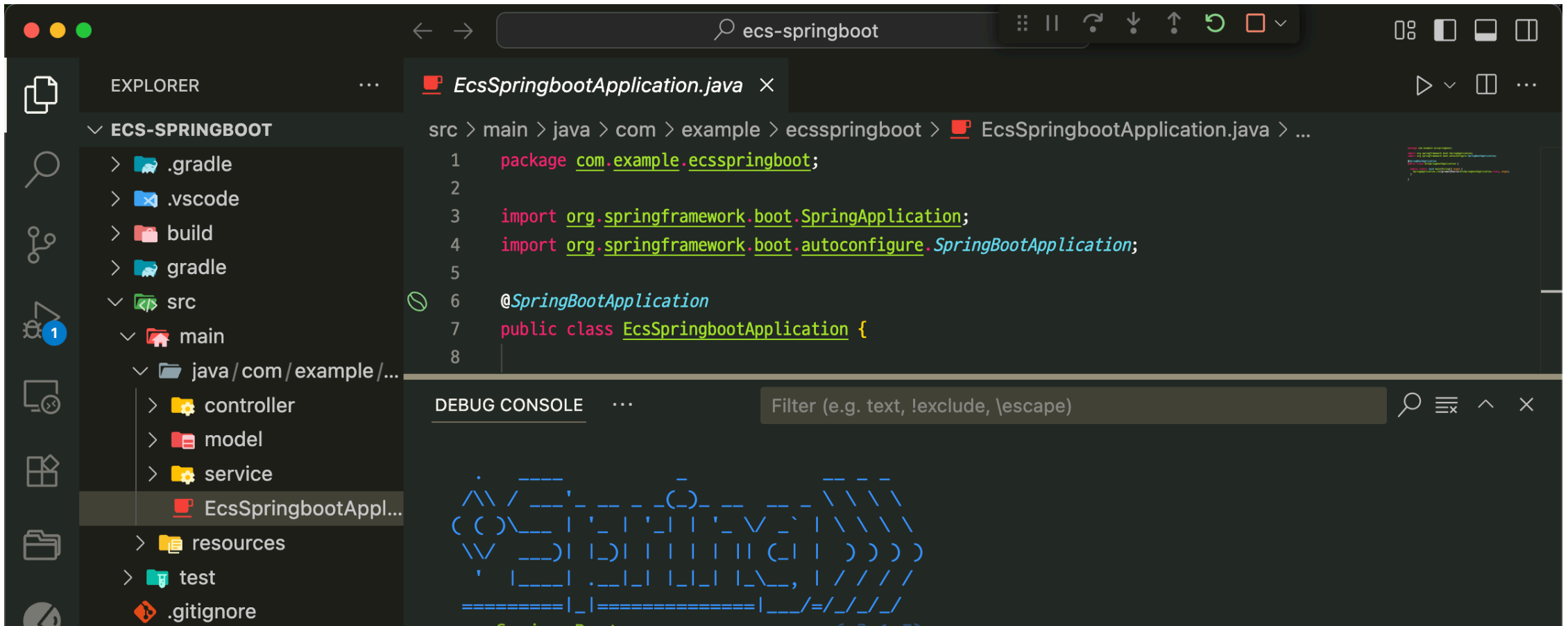
- Spring Boot는 Java 기반의 웹 애플리케이션을 빠르고 쉽게 개발할 수 있도록 도와주는 프레임워크입니다.



**Spring Boot**

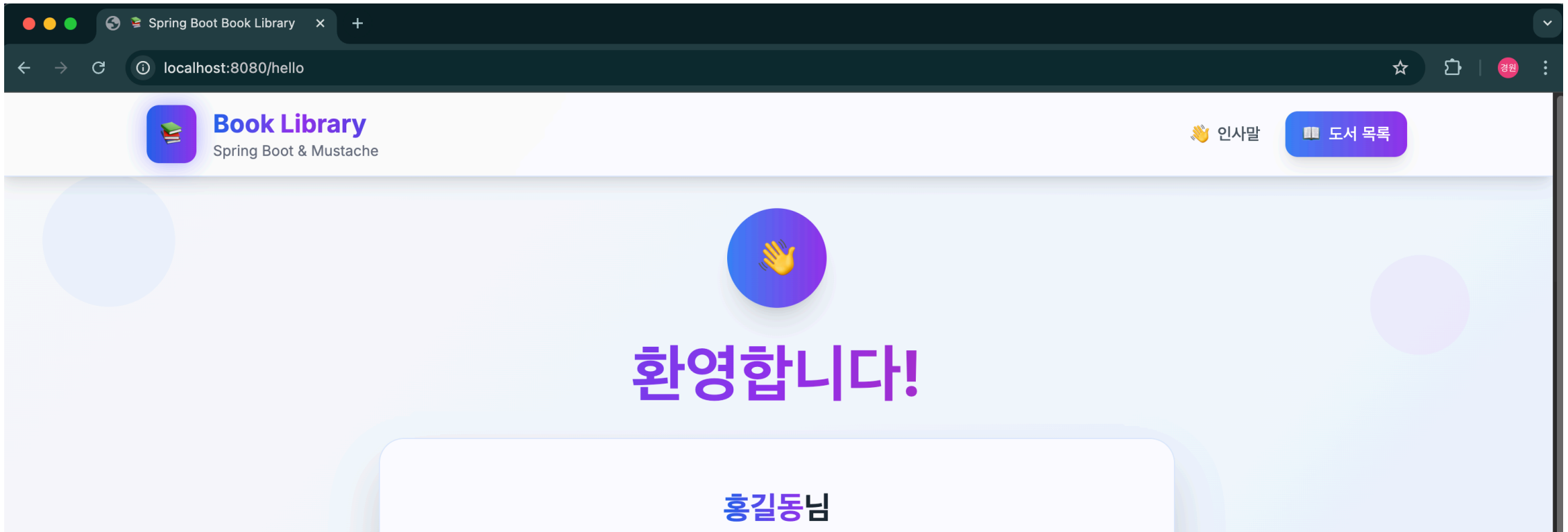
## 단계1: 프로젝트 실행





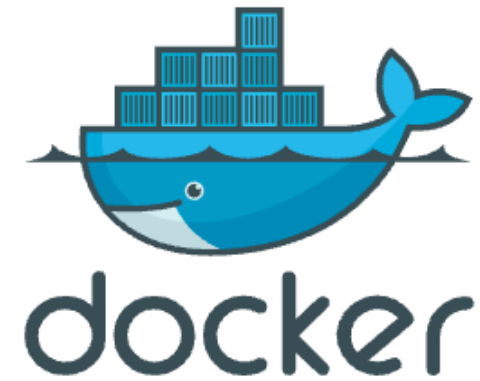
## 단계2: 프로젝트 테스트

http://localhost:8080/hello

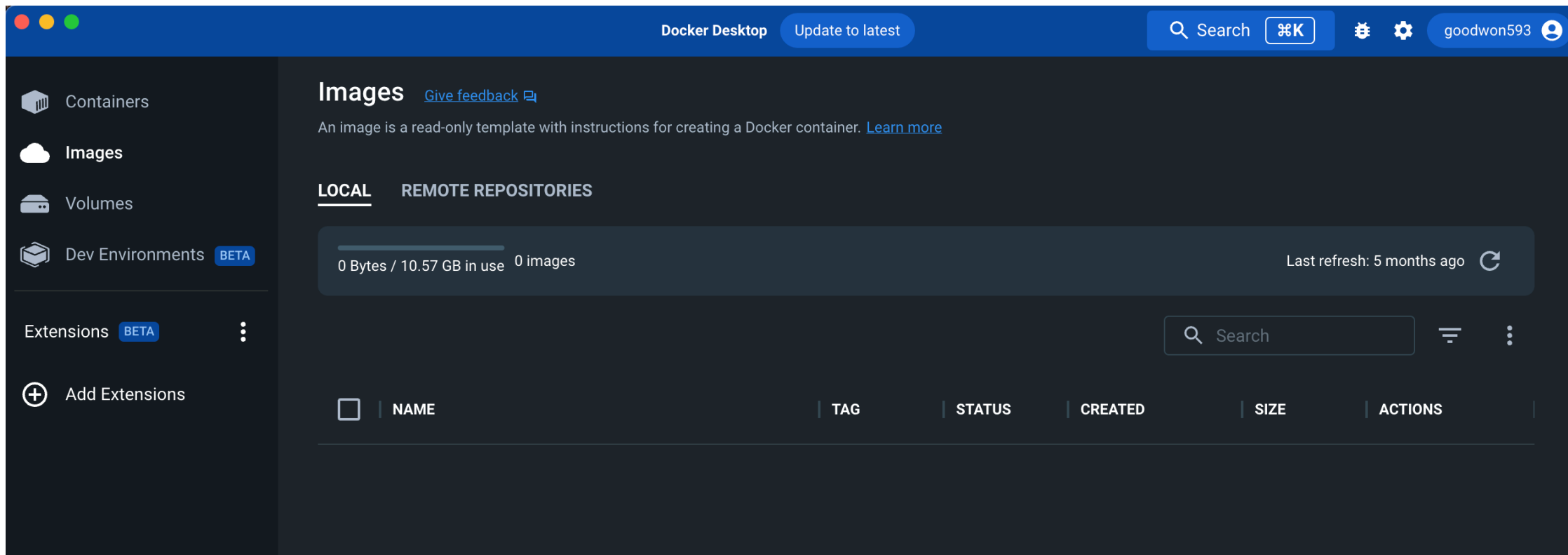


# Docker

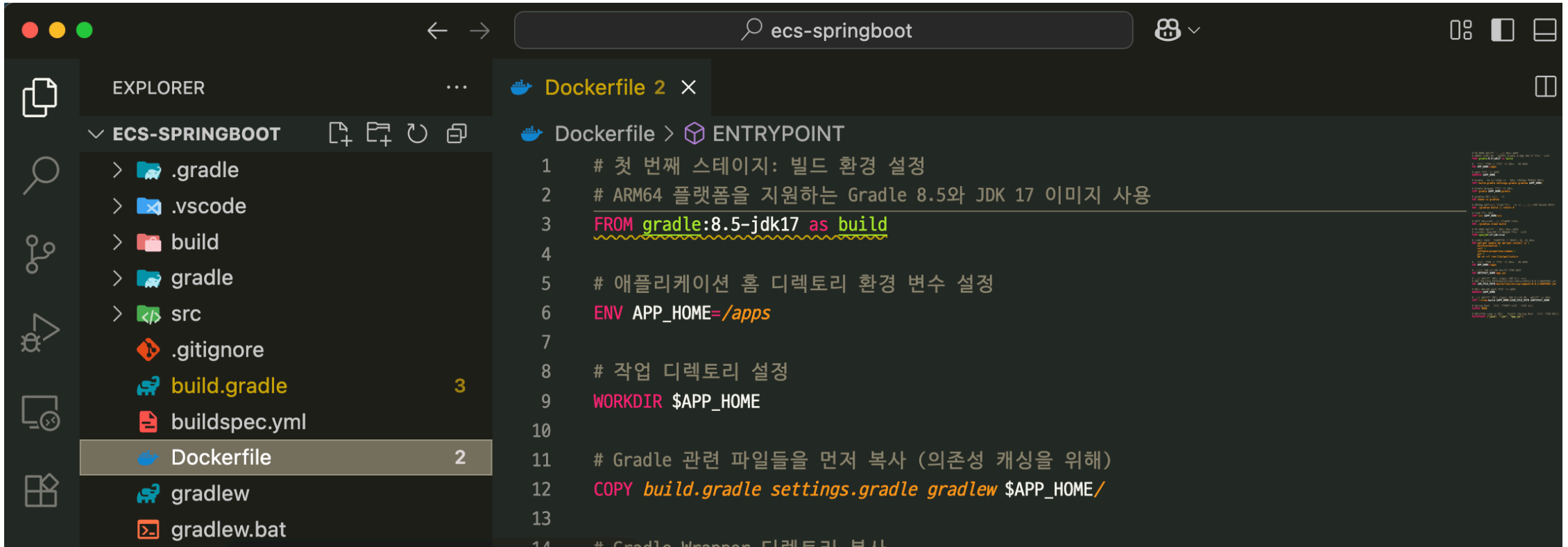
- Docker는 애플리케이션을 컨테이너(container)라는 가상화된 환경에서 실행할 수 있도록 해주는 오픈소스 플랫폼입니다.



# 단계1: Docker 실행



## 단계2: Dockerfile 확인



The screenshot shows the Visual Studio Code interface with a project named 'ecs-springboot'. The Explorer sidebar on the left shows the project structure, with 'Dockerfile' selected. The main editor displays the content of the Dockerfile, which is titled 'Dockerfile 2'. The Dockerfile contains the following instructions:

```
1 # 첫 번째 스테이지: 빌드 환경 설정
2 # ARM64 플랫폼을 지원하는 Gradle 8.5와 JDK 17 이미지 사용
3 FROM gradle:8.5-jdk17 as build
4
5 # 애플리케이션 홈 디렉토리 환경 변수 설정
6 ENV APP_HOME=/apps
7
8 # 작업 디렉토리 설정
9 WORKDIR $APP_HOME
10
11 # Gradle 관련 파일들을 먼저 복사 (의존성 캐싱을 위해)
12 COPY build.gradle settings.gradle gradlew $APP_HOME/
13
14 # Gradle Wrapper 디렉토리 복사
```



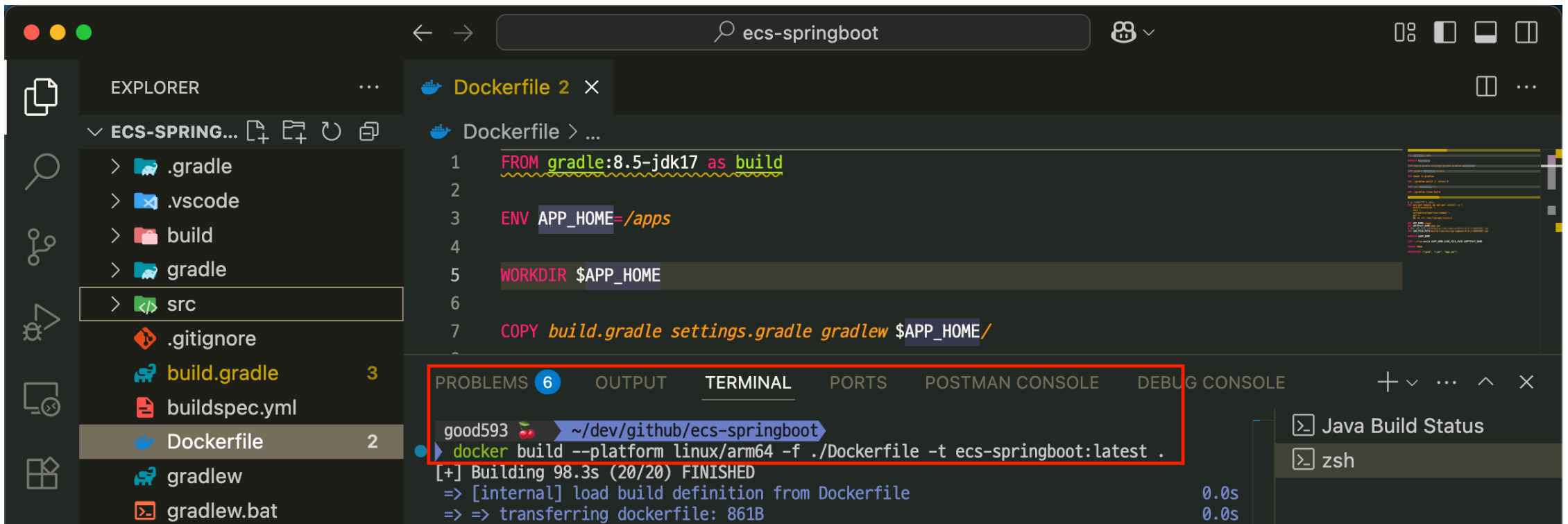
## 단계3: Docker image 생성

# docker image 생성

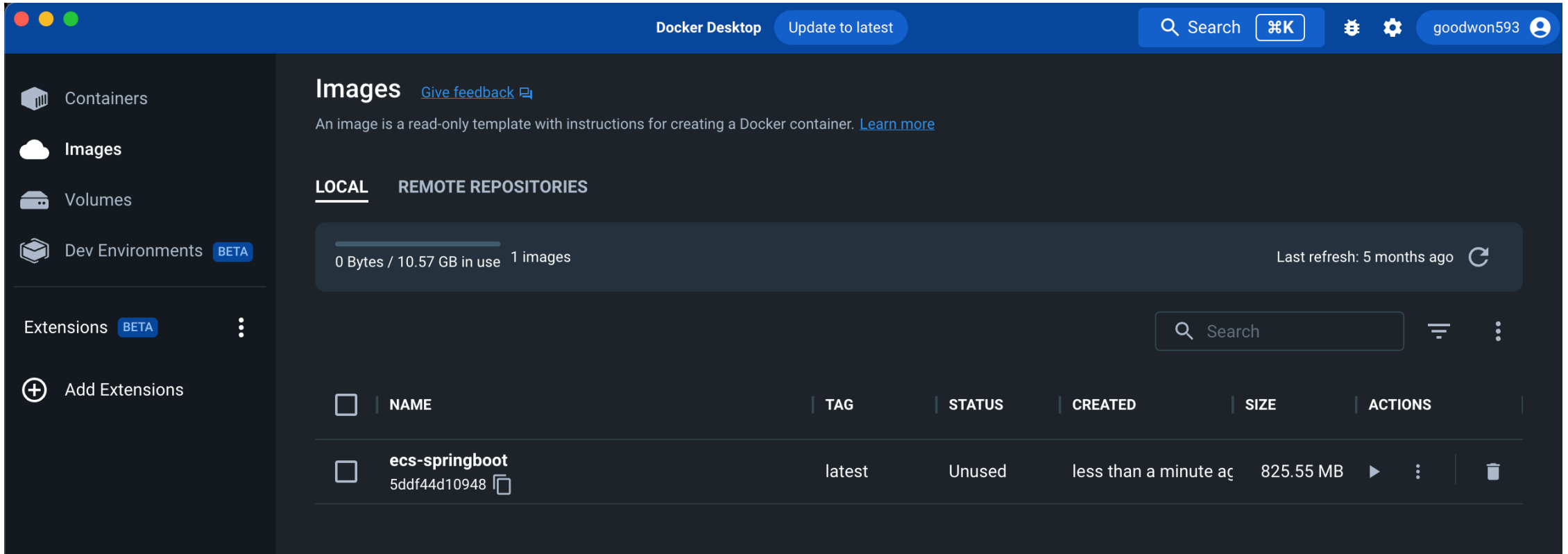
```
docker build -f ./Dockerfile -t ecs-springboot:latest .
```

# ARM 기반 CPU인 경우





```
docker build --platform linux/arm64 -f ./Dockerfile -t ecs-springboot:latest .
```



## 단계4: Docker image 생성 확인



The screenshot shows the Docker Desktop application window. The top bar is blue with the 'Docker Desktop' logo, an 'Update to latest' button, a search bar, and a user profile 'goodwon593'. The left sidebar contains navigation options: 'Containers', 'Images' (selected), 'Volumes', 'Dev Environments' (with a 'BETA' badge), 'Extensions' (with a 'BETA' badge), and 'Add Extensions'. The main area is titled 'Images' with a 'Give feedback' link. Below the title, it says 'An image is a read-only template with instructions for creating a Docker container. [Learn more](#)'. There are two tabs: 'LOCAL' (selected) and 'REMOTE REPOSITORIES'. A progress bar shows '0 Bytes / 10.57 GB in use' and '1 images'. A refresh button and 'Last refresh: 5 months ago' are on the right. Below this is a search bar and a table of images.

	NAME	TAG	STATUS	CREATED	SIZE	ACTIONS
<input type="checkbox"/>	<b>ecs-springboot</b> 5ddf44d10948 	latest	Unused	less than a minute ago	825.55 MB	  

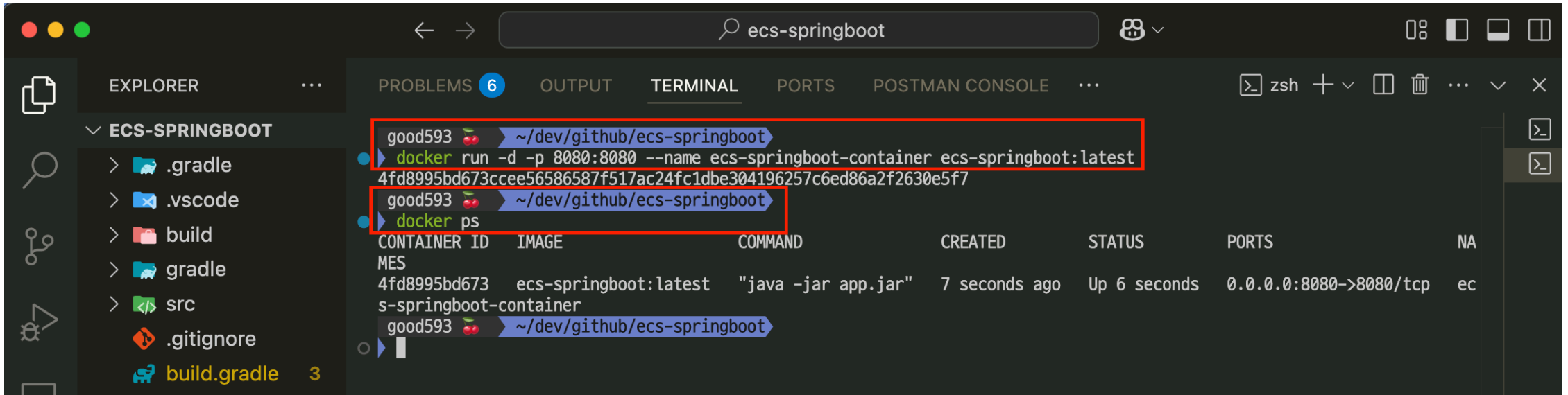
## 단계5: Docker Container 생성 및 실행

# Container 생성 및 실행

```
docker run -d -p 8080:8080 --name ecs-springboot-container ecs-springboot:latest
```

# 확인

```
docker ps
```



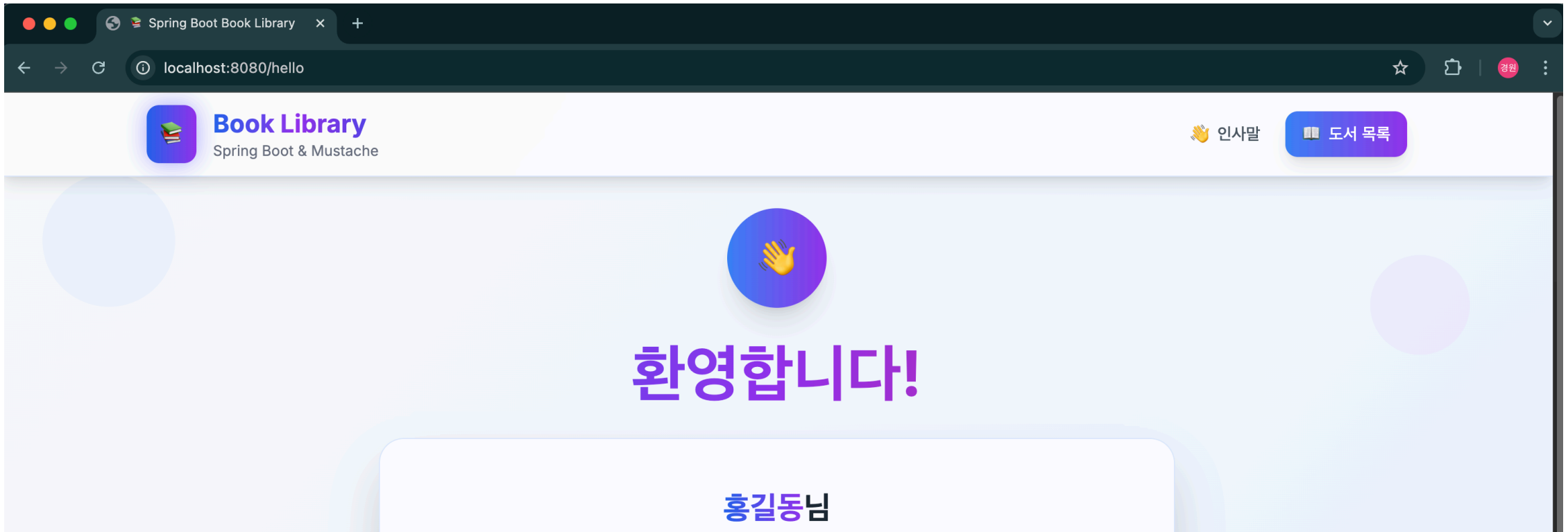
The screenshot shows the VS Code interface with the 'TERMINAL' tab active. The terminal output shows the successful execution of the 'docker run' command to create a container named 'ecs-springboot-container' and the subsequent 'docker ps' command which lists the running container. The container ID is 4fd8995bd673ccee56586587f517ac24fc1dbe304196257c6ed86a2f2630e5f7.

```
good593 ~/dev/github/ecs-springboot  
└─▶ docker run -d -p 8080:8080 --name ecs-springboot-container ecs-springboot:latest  
4fd8995bd673ccee56586587f517ac24fc1dbe304196257c6ed86a2f2630e5f7  
good593 ~/dev/github/ecs-springboot  
└─▶ docker ps
```

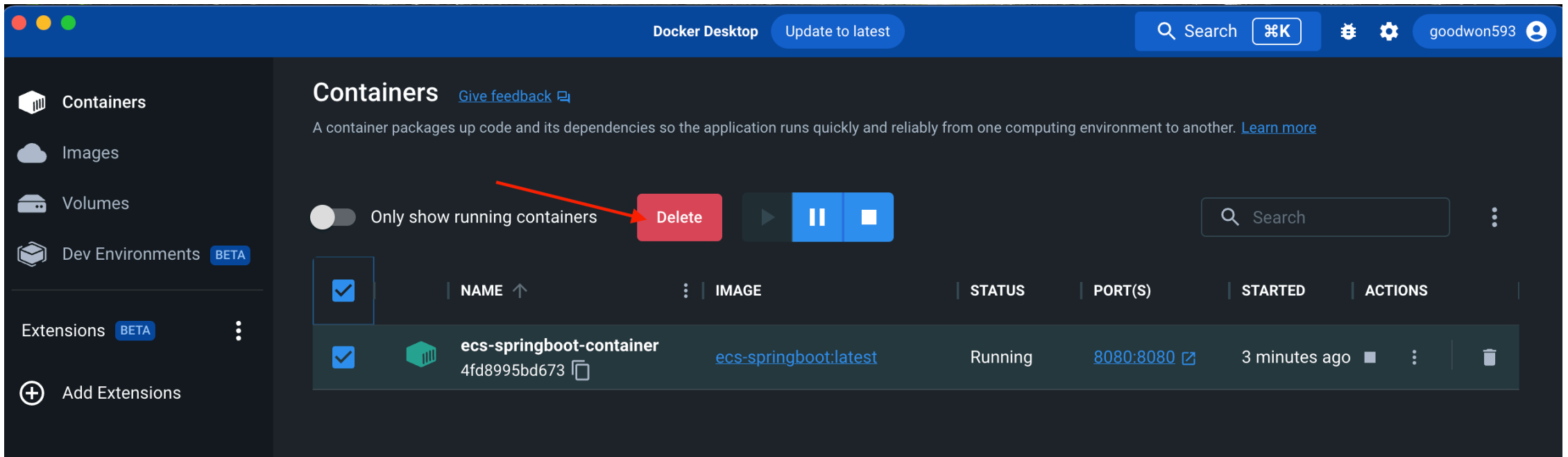
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NA
4fd8995bd673	ecs-springboot:latest	"java -jar app.jar"	7 seconds ago	Up 6 seconds	0.0.0.0:8080->8080/tcp	ec

## 단계6: 테스트

`http://localhost:8080/hello`






## 단계5: Docker Container 삭제









The screenshot shows the Docker Desktop application window. The top bar includes the 'Docker Desktop' title, an 'Update to latest' button, a search bar, and user information. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Extensions (marked BETA), and Add Extensions. The main area is titled 'Containers' and includes a 'Give feedback' link. Below the title is a description of containers and a 'Learn more' link. A toggle switch for 'Only show running containers' is present, with a red arrow pointing to a red 'Delete' button. Below this is a table of containers. The first container, 'ecs-springboot-container', is selected with a checkbox and has a trash icon in its actions column.

**Containers** [Give feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

☐ Only show running containers **Delete**   

☒ ☐ ☐

<input checked="" type="checkbox"/>	NAME ↑	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input checked="" type="checkbox"/>	 <b>ecs-springboot-container</b> 4fd8995bd673 	<a href="#">ecs-springboot:latest</a>	Running	<a href="#">8080:8080</a> 	3 minutes ago 	 

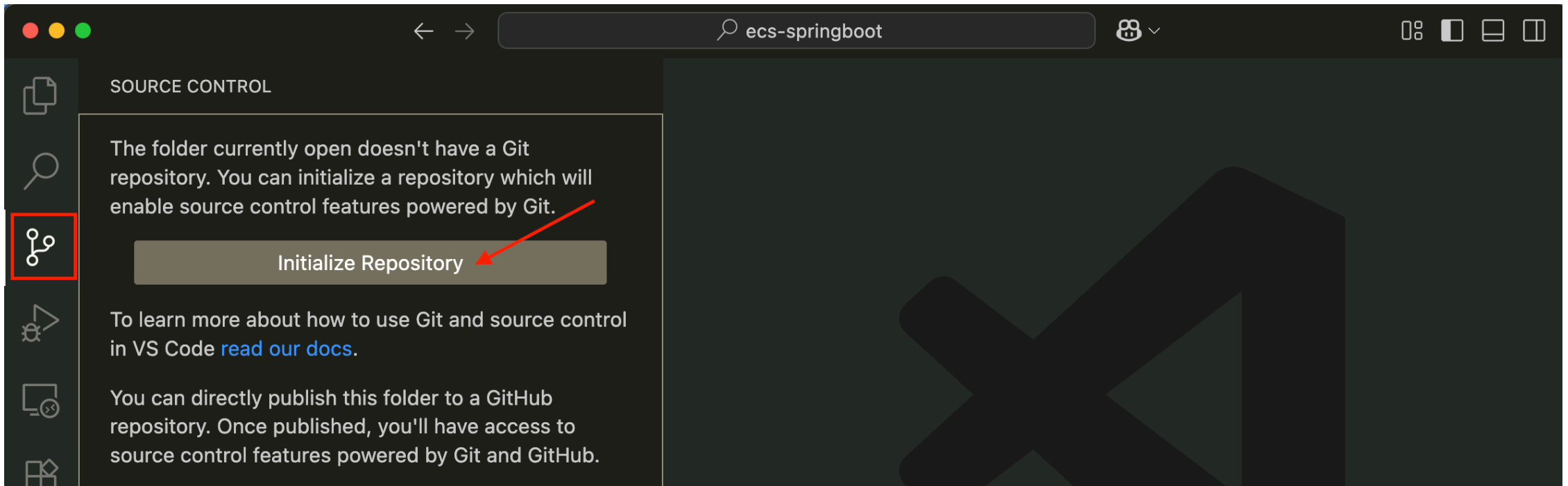
# Github

- GitHub는 개발자들이 코드를 저장하고, 협업하며, 버전 관리를 할 수 있는 웹 기반 플랫폼입니다.

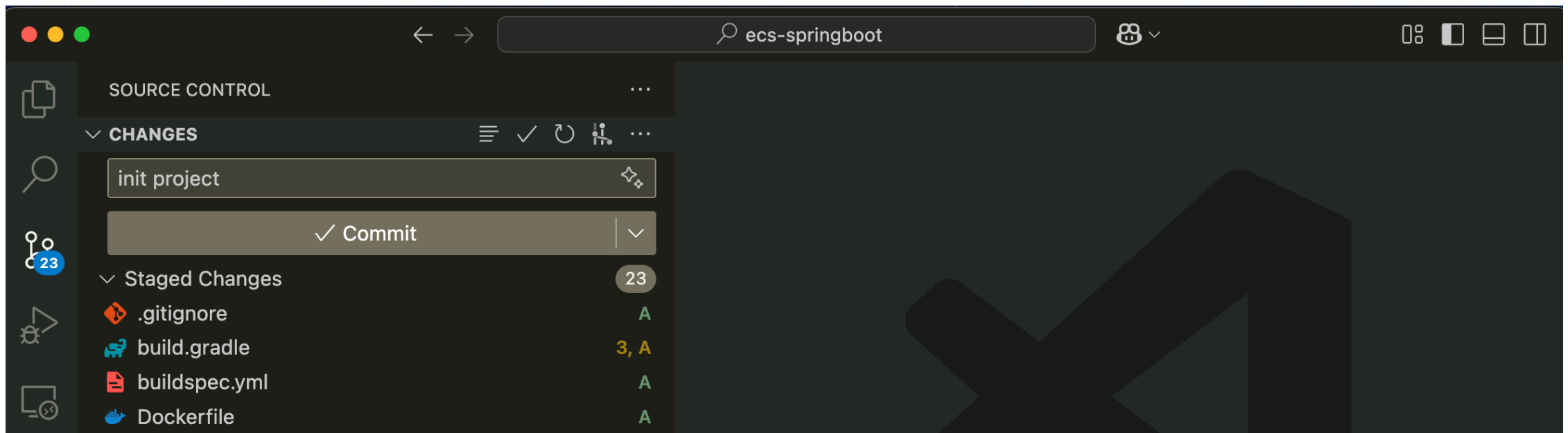


**GitHub Source**

## 단계1: Initialize Repository



## 단계2: Commit





## 단계3: Publish Branch

The screenshot shows the Source Control interface of an IDE. At the top, a search bar contains 'ecs-springboot'. Below it, a dropdown menu is open, showing two options: 'Publish to GitHub private repository' and 'Publish to GitHub public repository'. The second option is highlighted with a red box. A red arrow points from this option to the 'Publish Branch' button in the 'CHANGES' section. The 'CHANGES' section also contains a text input field with the placeholder 'Message (⌘Enter to commit on "main")' and a 'Publish Branch' button, which is also highlighted with a red box. On the right side, a table displays the commit history.

Graph	Description	Date	Author	Commit
	<b>main</b> init project	9 Jul 2025 ...	good593	a8f9f138

## 단계4: Github에서 확인

The screenshot shows a web browser window with the GitHub interface. The address bar displays 'github.com/good593/ecs-springboot'. The repository name 'good593 / ecs-springboot' is shown as 'Public'. Navigation tabs include 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. The 'Code' tab is active, showing a commit history table. The table lists three commits, all by 'good593' and labeled 'init project', each made '3 minutes ago'. The commit hashes are 'a8f9f13' and '3 minutes ago'. To the right, the 'About' section is partially visible, showing 'No description, website, or to provided.' and statistics for 'Readme', 'Activity', '0 stars', and '0 watching'.

Commit Hash	Author	Message	Time
a8f9f13	good593	init project	3 minutes ago
	good593	init project	3 minutes ago
	good593	init project	3 minutes ago