

영화 추천 프론트 엔드 다운로드(Docker)

DevOps 엔지니어의 업무인 배포를 실습합니다

1.영화 추천 프론트 엔드 다운로드

영화 추천 프론트 엔드 다운로드

https://github.com/gyeongnamit/movie_recommend_frontend 접속

The screenshot shows the GitHub repository page for `gyeongnamit/movie_recommend_frontend`. The repository is public and has 1 branch and 0 tags. The file list shows the following files and folders:

File/Folder	Commit Message	Time
public	프로젝트 공유	2 minutes ago
src	프로젝트 공유	2 minutes ago
Dockerfile	프로젝트 공유	2 minutes ago
README.md	프로젝트 공유	2 minutes ago
entrypoint.sh	프로젝트 공유	2 minutes ago
nginx.conf.template	프로젝트 공유	2 minutes ago
package.json	프로젝트 공유	2 minutes ago

The right sidebar shows the repository's metadata:

- About: No description, website, or topics provided.
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks
- Report repository
- Releases: No releases published
- Packages: No packages published

영화 추천 프론트 엔드 다운로드

The screenshot shows a web browser window displaying a GitHub repository page for 'gyeongnamit/movie_recommend_frontend'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted with a red box, and its dropdown menu is open, showing options to clone the repository using HTTPS, GitHub CLI, or the web URL. The 'Clone' option is selected, and the URL 'https://github.com/gyeongnamit/movie_recommend_frontend' is displayed. A red box highlights the 'Copy url to clipboard' button. The repository's file structure is listed on the left, including 'public', 'src', 'Dockerfile', 'README.md', 'entrypoint.sh', 'nginx.conf.template', and 'package.json'. The right sidebar shows the 'About' section with no description, website, or topics provided, and the 'Releases' and 'Packages' sections with no published items.

GitHub - gyeongnamit/movie_rec x +

https://github.com/gyeongnamit/movie_recommend_frontend

Product Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

gyeongnamit / movie_recommend_frontend Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

gyeongnamit 프로젝트 공유

public	프로젝트 공유
src	프로젝트 공유
Dockerfile	프로젝트 공유
README.md	프로젝트 공유
entrypoint.sh	프로젝트 공유
nginx.conf.template	프로젝트 공유
package.json	프로젝트 공유

2 minutes ago

2 minutes ago

Clone

HTTPS GitHub CLI

Copy url to clipboard

https://github.com/gyeongnamit/movie_recommend_frontend

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Report repository

Releases

No releases published

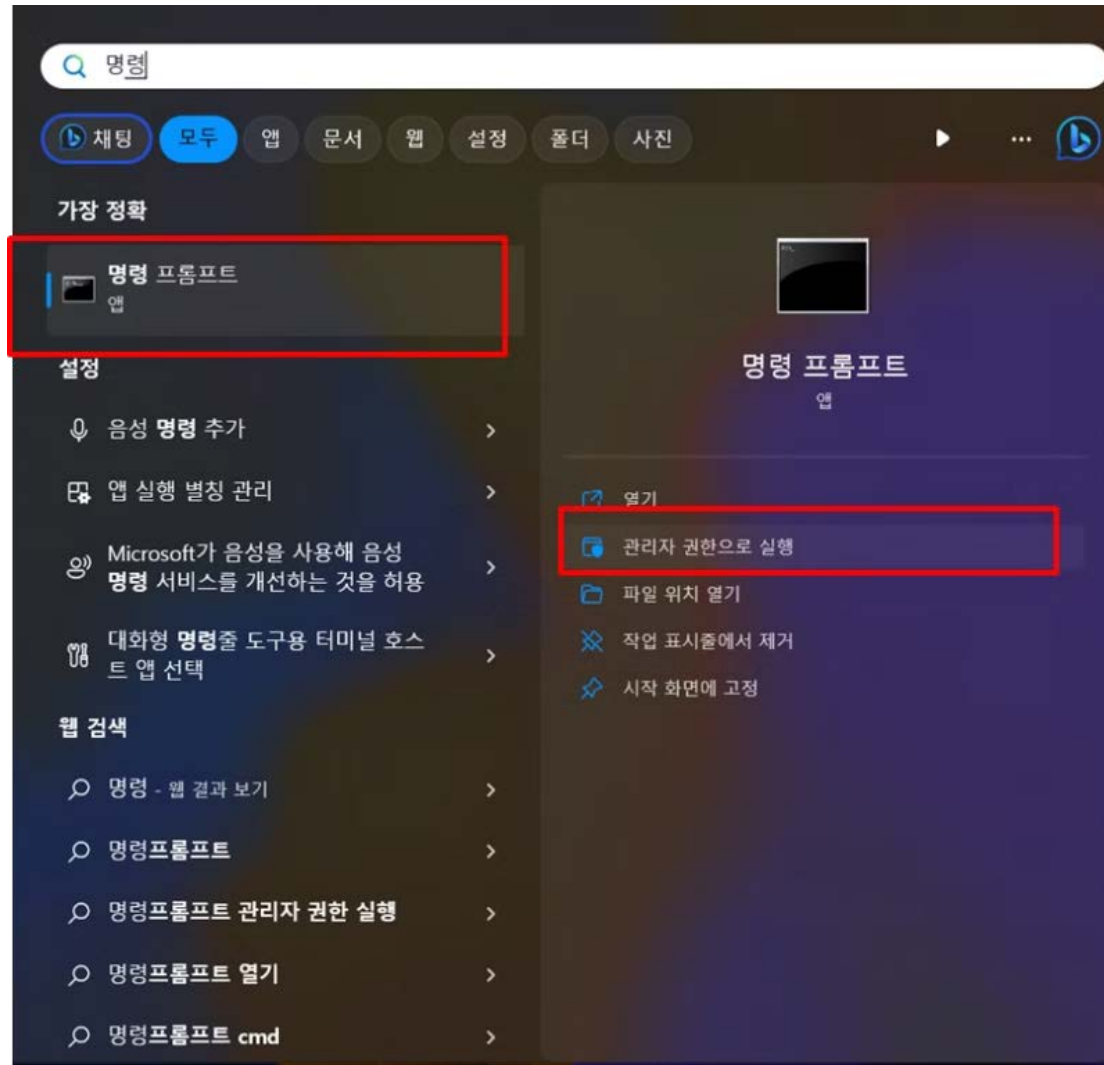
Packages

No packages published

README


영화 추천 프론트 엔드 다운로드

명령 프롬프트를 관리자 권한으로 실행



영화 추천 프론트 엔드 다운로드

영화 추천 프로젝트 생성

 관리자: 명령 프롬프트

```
C:\WINDOWS\system32>cd c:\movie_project01
```

프로젝트를 저장할 폴더로 이동

```
c:\movie_project01>git clone https://github.com/gyeongnamit/movie_recommend_frontend.git
```

프로젝트 다운로드

2.Docker Image (설계도) 만들고 Docker hub (도서관) 에 등록

도커 이미지를 만든 후 도커 허브에 등록하는 과정입니다.

2.1 Docker Hub 회원명 확인

Docker Hub는 Docker Image(식당 설계도)를 저장하고 공유하는 온라인 저장소입니다

Docker Hub

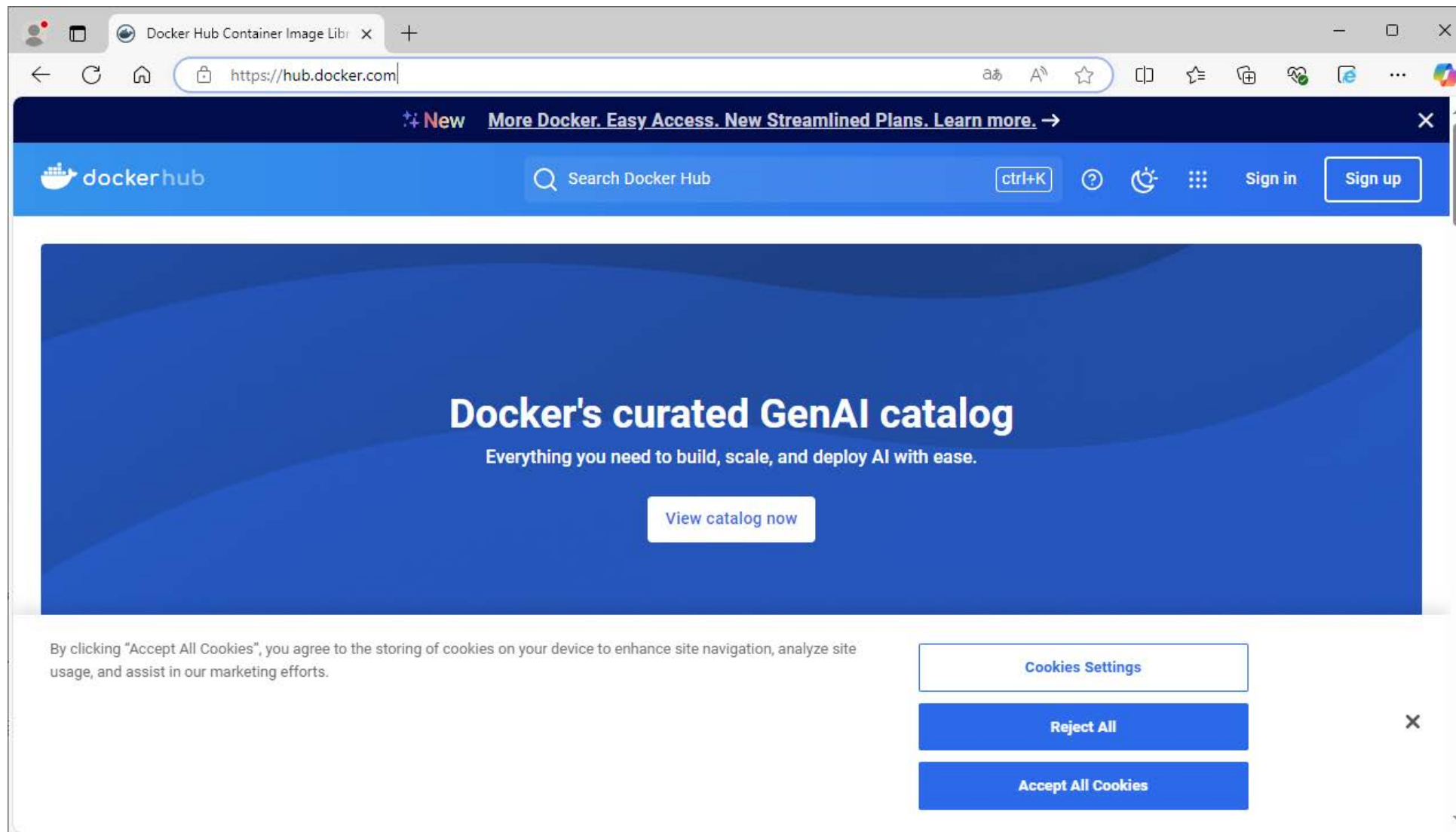
Docker Hub는 Docker Image(식당 설계도)를 저장하고 공유하는 온라인 저장소입니다. 개발자들은 자신이 만든 이미지를 여기에 올려 공유하고, 다른 이들은 필요한 이미지를 가져다 사용할 수 있습니다.

이번 프로젝트에서는 영화 추천 프론트 엔드를 Docker Image로 만들어 Docker Hub에 저장합니다.

- **설계도 공유 및 재사용:** 누구나 쉽게 Docker Image를 사용할 수 있습니다.
- **공식 이미지 제공:** Java, Python, Node.js 등 다양한 프로그래밍 환경의 공식 이미지를 제공합니다.
- **무료 저장 및 다운로드:** Docker Hub는 이미지 저장 및 다운로드를 무료로 제공합니다.

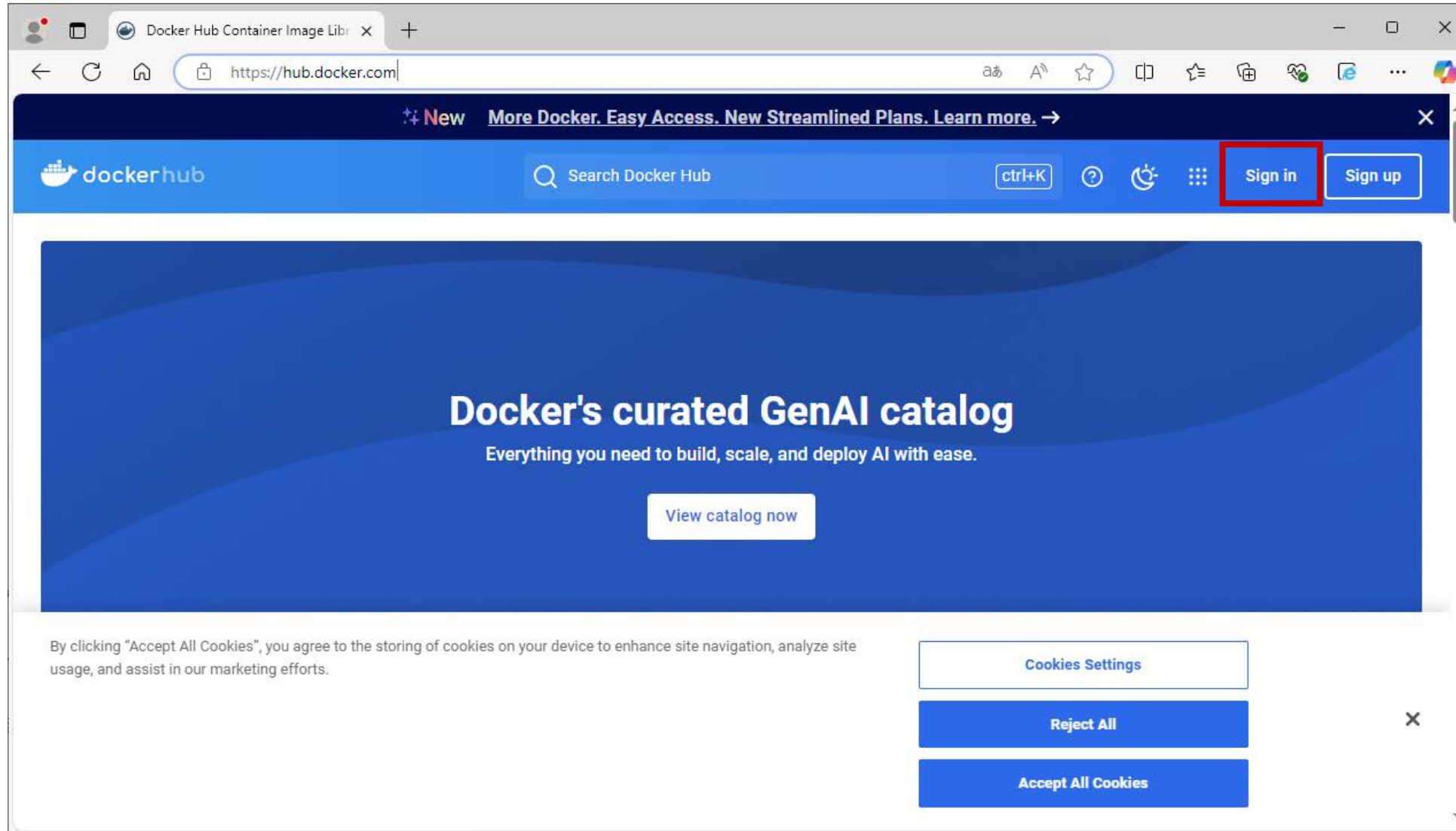
Docker Hub 회원명 확인

<https://hub.docker.com/> 접속 합니다



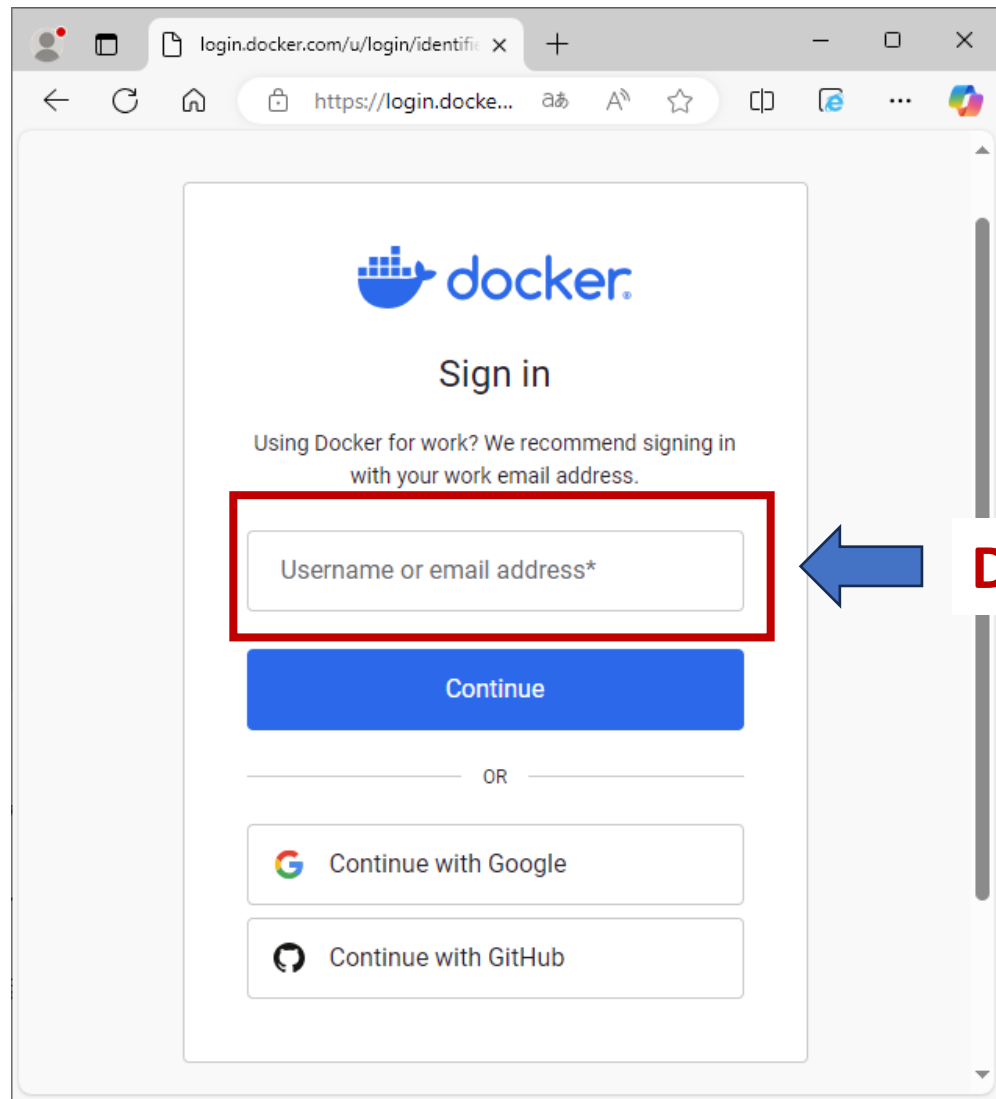
Docker Hub 회원명 확인

sign in 을 클릭 합니다




Docker Hub 회원명 확인

Docker Hub에 가입한 이메일을 입력 합니다



login.docker.com/u/login/identifi

https://login.docke...

 docker


Sign in


Using Docker for work? We recommend signing in with your work email address.

Username or email address*

Continue

OR

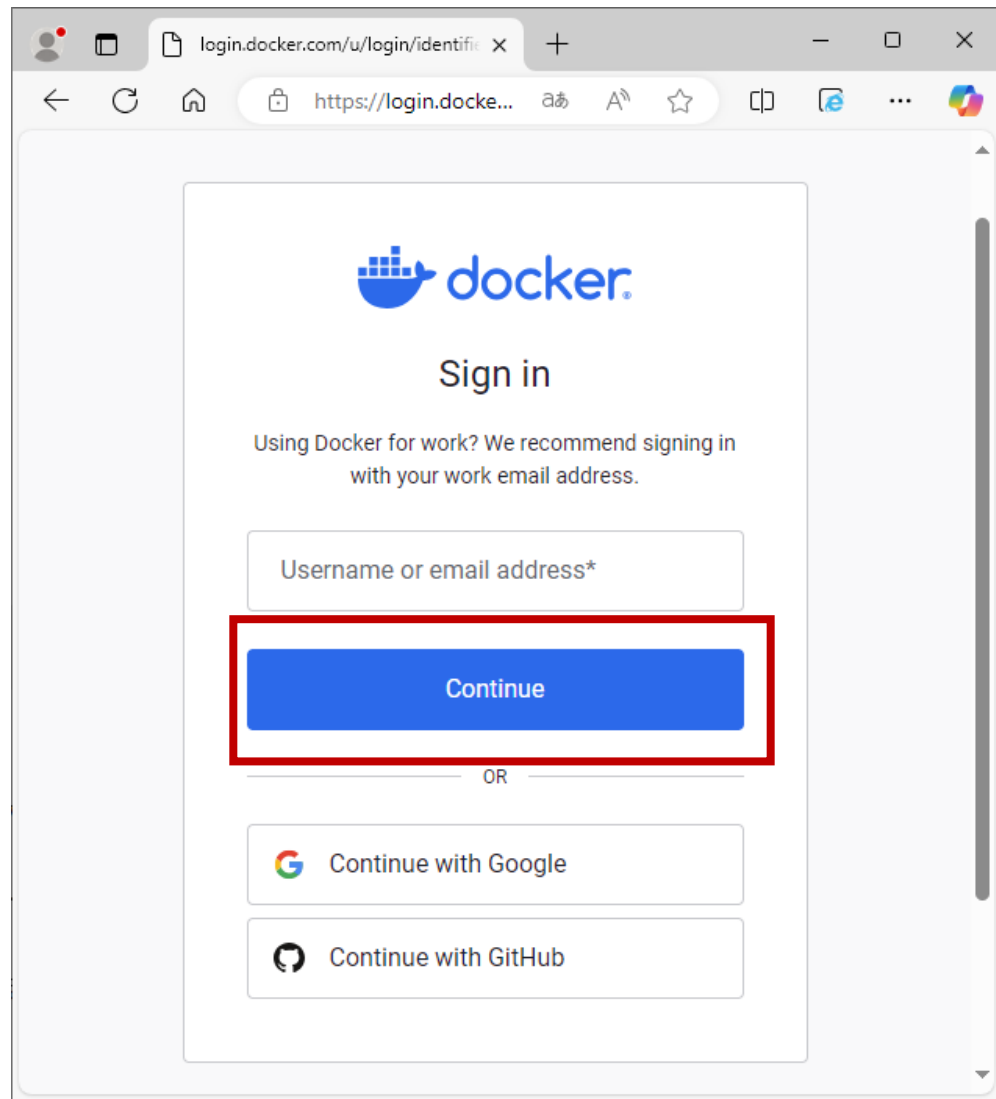
 Continue with Google

 Continue with GitHub

Docker Hub에 가입한 이메일을 입력합니다

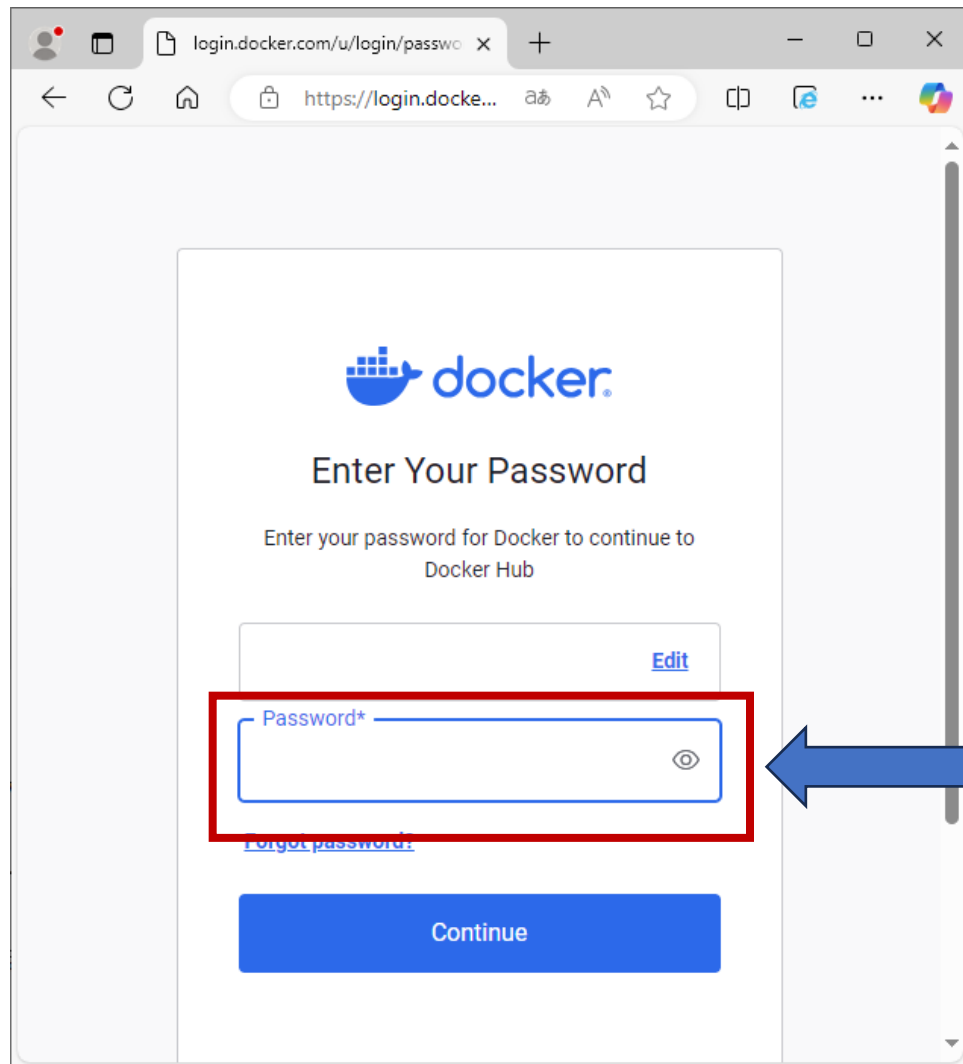
Docker Hub 회원명 확인

Continue를 클릭 합니다



Docker Hub 회원명 확인

Docker Hub에 가입한 비밀번호를 입력 합니다

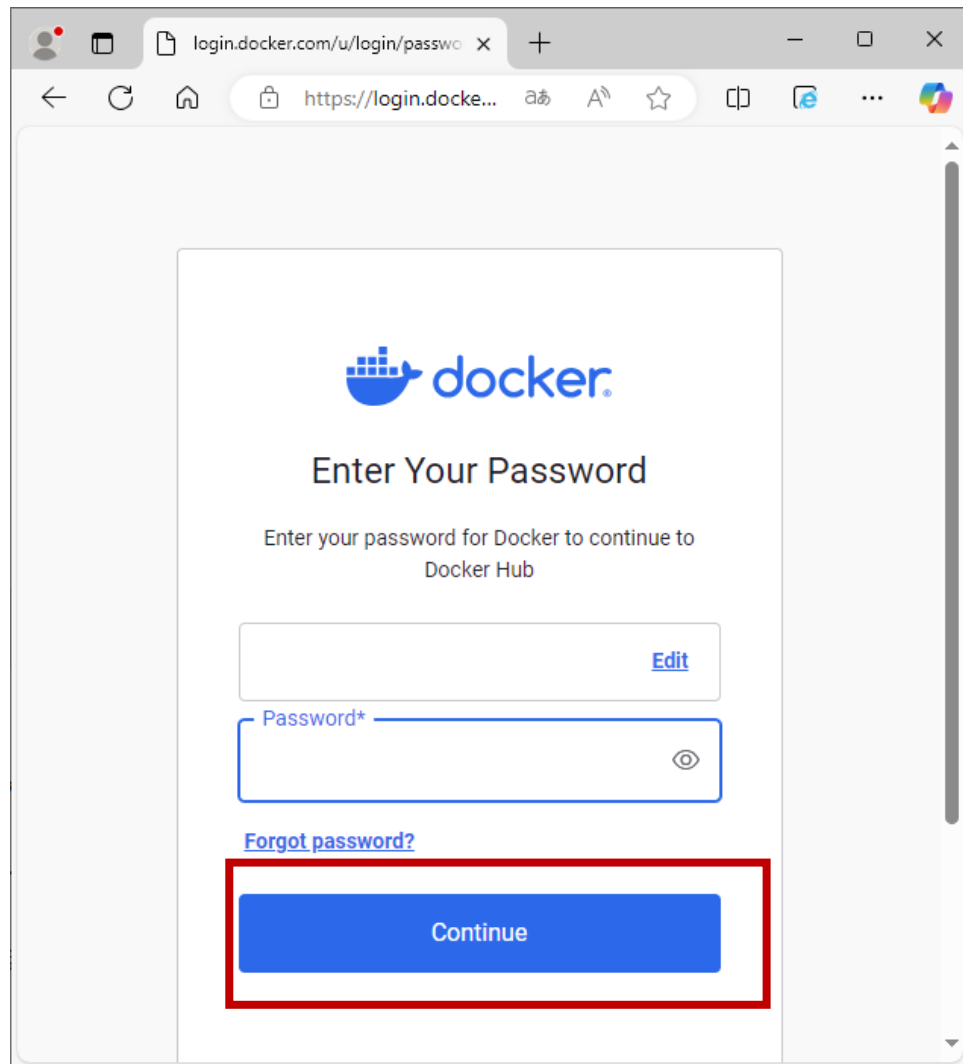


The screenshot shows the Docker Hub login page in a web browser. The page has the Docker logo at the top, followed by the text "Enter Your Password" and "Enter your password for Docker to continue to Docker Hub". Below this is a password input field labeled "Password*" with a red rectangular box around it. To the right of the input field is a blue arrow pointing left towards the text "Docker Hub에 가입한 비밀번호를 입력합니다". Below the input field is a "Continue" button. There are also links for "Edit" and "Forgot password?" near the input field.

Docker Hub에 가입한 비밀번호를 입력합니다

Docker Hub 회원명 확인

Continue 버튼을 클릭 합니다



The screenshot shows a web browser window with the URL `login.docker.com/u/login/passwo`. The page displays the Docker logo and the heading "Enter Your Password". Below this, it says "Enter your password for Docker to continue to Docker Hub". There is a password input field with an "Edit" link to its right. Below the password field is a "Forgot password?" link. At the bottom, a blue "Continue" button is highlighted with a red rectangular border.

[Edit](#)

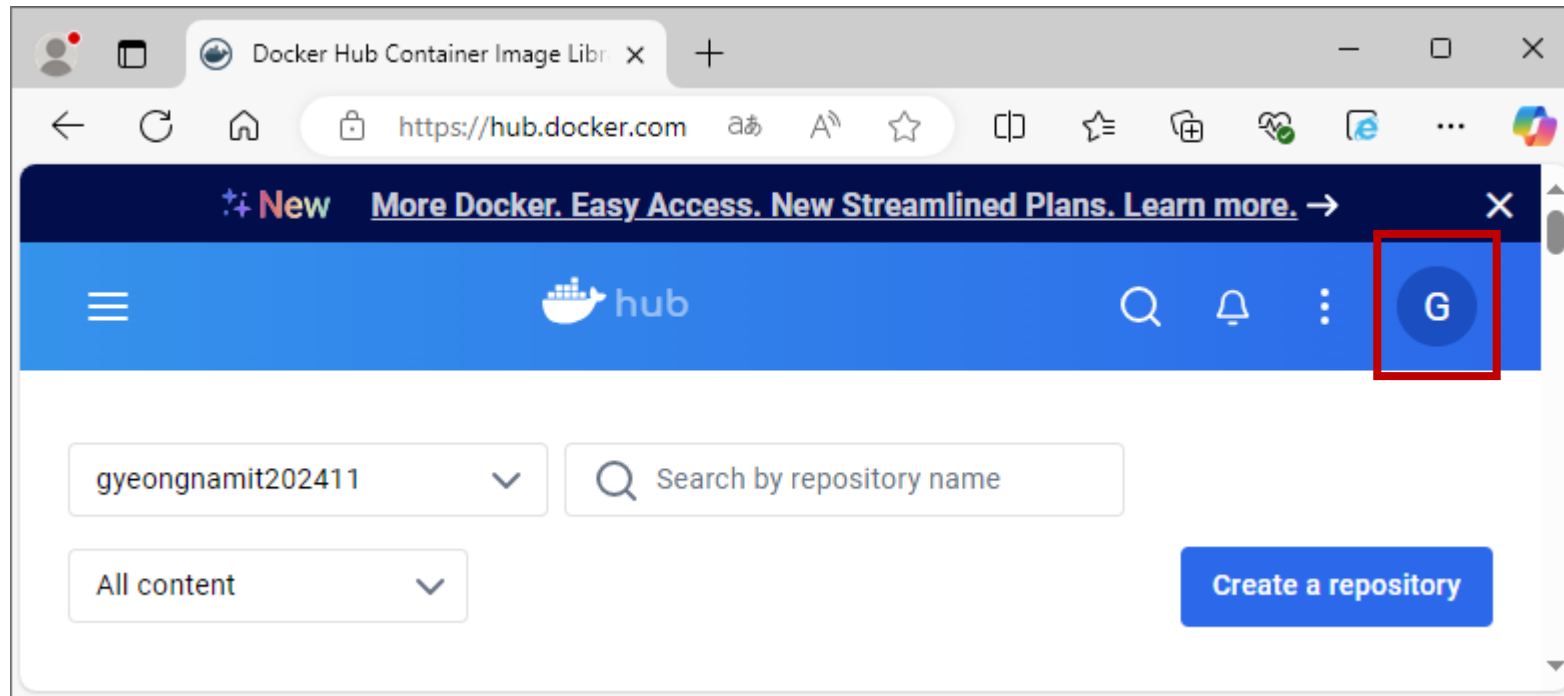
Password*

[Forgot password?](#)

Continue

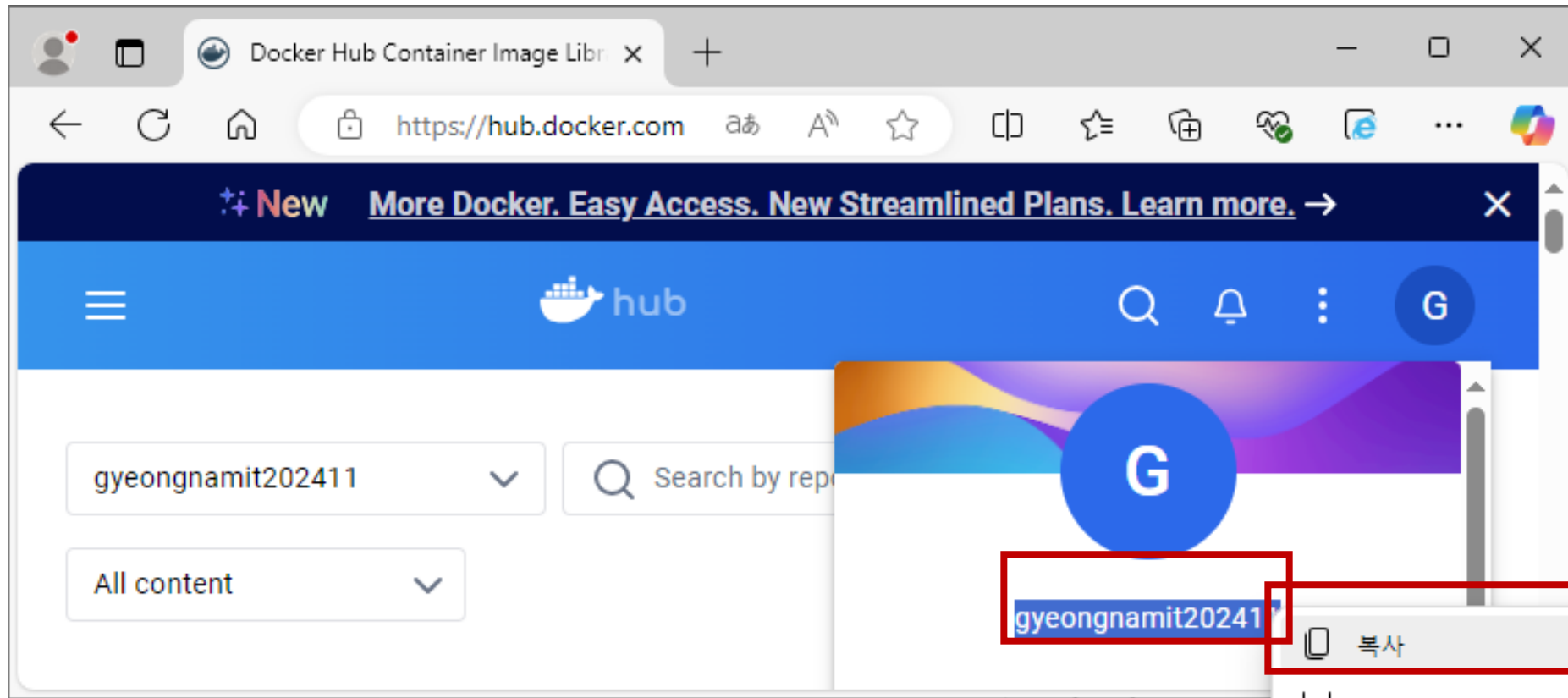
Docker Hub 회원명 확인

화면 오른쪽의 사용자명 첫글자 아이콘을 클릭 합니다



Docker Hub 회원명 확인

사용자 이름을 드래그 한 후 마우스 오른쪽 버튼을 클릭해서 복사 합니다. 복사한 사용자 이름을 메모장에 붙여 넣은 후 저장 합니다



2.2.도커 이미지 만들기

이제 실제 영화 추천 프론트 엔드를 위한 Docker Image를 만들어 보겠습니다.

Dockerfile (식당 설계 스케치)을 사용하여 Docker Image (실제 식당 설계도)를 생성합니다.

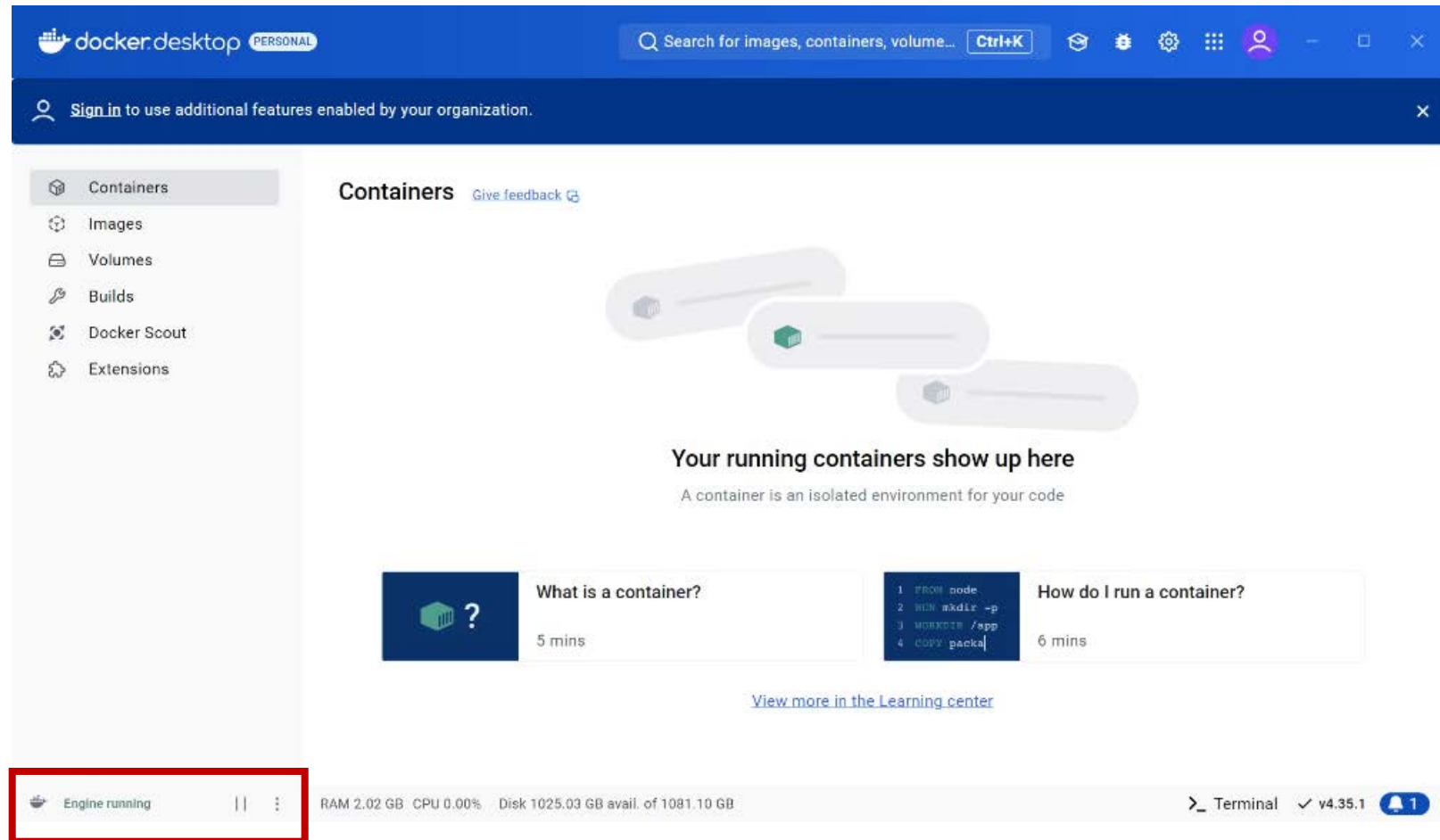
Docker 실행

바탕화면에서 Docker Desktop 아이콘 더블클릭



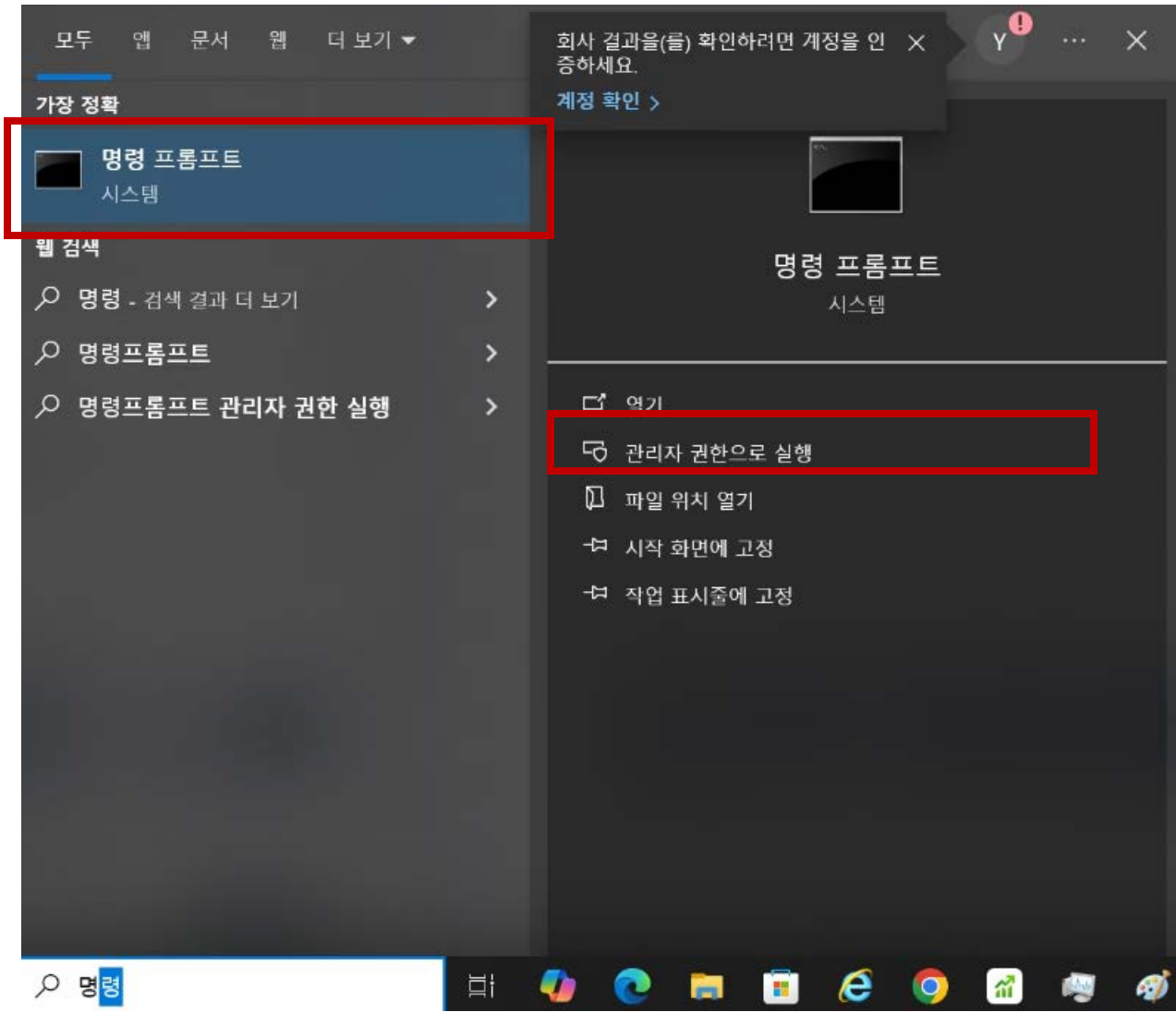
Docker 실행

Engine running 메시지가 나올때 까지 때까지 기다리세요




도커 이미지 만들기

명령 프롬프트를 관리자 권한으로 실행 합니다



도커 이미지 만들기

영화 추천 프론트 엔드 프로그램을 구현한 C:\movie_project01\movie_recommend_frontend 로 이동합니다.

 관리자: 명령 프롬프트

```
c:\movie_project01>cd c:\movie_project01
```


```
c:\movie_project01>cd movie_recommend_frontend
```

```
c:\movie_project01\movie_recommend_frontend>■
```

도커 이미지 만들기

Docker Hub에 로그인 합니다.

```
bash
```


 Copy code

```
docker login -u <사용자 이름>
```

DockerHub에 가입한 사용자 이름을 입력합니다.

**gyeongnamit202411 대신에 17페이지
에서 확인한 Docker Hub 사용자 이름을
입력합니다**



 관리자: 명령 프롬프트

```
c:\movie_project01\movie_recommend_frontend>docker login -u gyeongnamit202411
```


Password:  **Docker Hub에
가입할때 입력한 비밀번호 입력**
Login Succeeded

```
c:\movie_project01\movie_recommend_frontend>
```

도커 이미지 만들기

`docker build -t gyeongnamit202411/movie_recommend_frontend .`


bash

 Copy code

```
docker build -t gyeongnamit202411/movie_recommend_frontend .
```




gyeongnamit202411 대신에 17페이지에서 확인한 Docker Hub 사용자 이름을 입력합니다

 관리자: 명령 프롬프트 - `docker build -t gyeongnamit202411/movie_recommend_frontend .`

```
c:\movie_project01\movie_recommend_frontend>docker build -t gyeongnamit202411/movie_recommend_frontend .
```


도커 이미지 만들기

에러 없이 이미지가 생성 되었는지 확인

 관리자: 명령 프롬프트

```
c:\movie_project01\movie_recommend_frontend>docker build -t gyeongnamit202411/movie_recommend_frontend .
[+] Building 1.2s (19/19) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.57kB
=> [internal] load metadata for docker.io/library/nginx:1.23-alpine
=> [internal] load metadata for docker.io/library/node:16
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [builder 1/6] FROM docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c6389705c0b
=> => resolve docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c6389705c0b
=> [stage-1 1/7] FROM docker.io/library/nginx:1.23-alpine@sha256:02ffd439b71d9ea9408e449b568f65c0bbbb94bebd8750f1d80231ab6496008e
=> => resolve docker.io/library/nginx:1.23-alpine@sha256:02ffd439b71d9ea9408e449b568f65c0bbbb94bebd8750f1d80231ab6496008e
=> [internal] load build context
=> => transferring context: 2.80kB
=> CACHED [stage-1 2/7] WORKDIR /etc/nginx/templates
=> CACHED [stage-1 3/7] RUN apk update && apk add --no-cache bash gettext && rm -rf /var/cache/apk/*
=> CACHED [builder 2/6] WORKDIR /app
=> CACHED [builder 3/6] COPY package.json yarn.lock ./
=> CACHED [builder 4/6] RUN yarn install && yarn add axios
=> CACHED [builder 5/6] COPY . .
=> CACHED [builder 6/6] RUN yarn build
=> CACHED [stage-1 4/7] COPY --from=builder /app/build /usr/share/nginx/html
=> CACHED [stage-1 5/7] COPY nginx.conf.template /etc/nginx/templates/nginx.conf.template
=> CACHED [stage-1 6/7] COPY entrypoint.sh /entrypoint.sh
=> CACHED [stage-1 7/7] RUN chmod +x /entrypoint.sh
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:920b38bc9a3792338145e9af9177d3a13f44bdf00ffbb2328963d9aa529aa09a
=> => exporting config sha256:8387077a2fac6b1c0186174ad4df46abf3d3e1510ad2cd9441eb8a97e64c695c
=> => exporting attestation manifest sha256:7f738018cbe68de622a991551f77a5dd0f63b54995908b1bf5de35fdca66c6aa
=> => exporting manifest list sha256:ad3ecc9357d94002e9a97dba4c52a99a6a128d34b0c6eb4aa2abcfbcca455e63
=> => naming to docker.io/gyeongnamit202411/movie_recommend_frontend:latest
=> => unpacking to docker.io/gyeongnamit202411/movie_recommend_frontend:latest
```

```
c:\movie_project01\movie_recommend_frontend>_
```

에러 없이
이미지가 생성되는지
확인

```
bash
```

[Copy code](#)

```
docker build -t gyeongnamit202411/movie_recommend_frontend .
```

이 명령어는 Docker(도커)라는 도구를 이용해 프로그램이 실행되는 환경(이미지)을 만드는 작업을 수행합니다.

쉽게 말하면, 햄버거를 만들기 위한 레시피(Dockerfile)를 사용해서 포장된 완성품(도커 이미지)을 만드는 과정입니다.

- **docker build**

- 도커에서 새로운 이미지(환경)를 만드는 명령어입니다.
- 예: 요리법(Dockerfile)을 따라 요리를 조리하는 과정입니다.

- **-t (태그 이름 지정)**

- -t 옵션은 새로 만든 이미지에 이름과 태그를 붙이는 역할을 합니다.
- 예: 만든 요리에 이름표를 붙이는 것과 같습니다.
- 여기서 gyeongnamit202411/movie_recommend_frontend는 도커 이미지 이름입니다.
 - gyeongnamit202411 → 사용자 이름(요리사 이름)
 - movie_recommend_frontend → 프로그램 이름(요리 메뉴 이름)

- **.(점)**

- 현재 폴더에 있는 Dockerfile(레시피)를 사용하겠다는 의미입니다.
- 예: 요리를 만들기 위해 현재 주방에서 재료와 도구를 가져옵니다.

도커 이미지 만들기 설명

- **요리법(Dockerfile) 확인**
 - 현재 폴더에서 Dockerfile을 찾습니다.이 파일에는 프로그램을 설치하고 실행하는 방법이 적혀 있습니다.
- **재료 준비 및 설치**
 - Dockerfile에 따라 프로그램에 필요한 재료(파일)와 도구를 복사합니다.
 - 예: 재료를 씻고, 손질하고, 요리를 시작합니다.
- **환경 세팅 및 포장(이미지 생성)**
 - 프로그램을 실행할 수 있는 환경을 설정합니다.
 - 설정이 끝나면 포장된 프로그램 파일(이미지)을 생성합니다.
 - 예: 포장된 햄버거를 진열할 준비를 마칩니다.

도커 이미지 만들기 설명

- 이 명령어는 햄버거 조리 과정과 비슷합니다.
- **레시피(Dockerfile) 확인**
 - 어떤 재료와 도구가 필요한지 살펴봅니다.
- **재료 준비 및 조리**
 - 필요한 재료(파일)와 도구(패키지)를 조립하고, 요리를 완성합니다.
- **포장 완료(이미지 생성)**
 - 완성된 햄버거(이미지)를 포장해서 판매 준비를 완료합니다.
- **이름표 붙이기**
 - 햄버거에 이름(movie_recommend_frontend)을 붙여서 구분합니다.


2.3.도커 이미지 DockerHub 등록

Docker Hub는 개발자들이 자신의 Docker Image를 쉽게 관리하고 다른 사람들과 협업할 수 있는 플랫폼을 제공합니다. 이는 마치 여러 식당의 설계도를 한 곳에 모아둔 거대한 도서관과 같습니다.

도커 이미지 DockerHub 등록

Docker Hub에 이미지 업로드


bash

 Copy code

```
docker push gyeongnamit202411/movie_recommend_frontend
```




gyeongnamit202411 대신에 17페이지에서 확인한 Docker Hub 사용자 이름을 입력합니다

 관리자: 명령 프롬프트

```
c:\movie_project01\movie_recommend_frontend>docker push gyeongnamit202411/movie_recommend_frontend_
```

도커 이미지 DockerHub 등록

 관리자: 명령 프롬프트

```
c:\movie_project01\movie_recommend_frontend>docker push gyeongnamit202411/movie_recommend_frontend
Using default tag: latest
The push refers to repository [docker.io/gyeongnamit202411/movie_recommend_frontend]
595a9495cfe6: Pushed
c3fbff4f43c6: Pushed
62abd1912d34: Pushed
4f4fb700ef54: Layer already exists
c28dd04bc1fd: Layer already exists
f56be85fc22e: Layer already exists
a20276ab5ec8: Layer already exists
59b9d2200e63: Layer already exists
3e1e579c95fe: Layer already exists
547a97583f72: Layer already exists
c23b4f8cf279: Layer already exists
31cdf5c1852b: Pushed
1f21f983520d: Layer already exists
2ce963c369bc: Layer already exists
latest: digest: sha256:ad3ecc9357d94002e9a97dba4c52a99a6a128d34b0c6eb4aa2abcfbcca455e63 size: 856

c:\movie_project01\movie_recommend_frontend>_
```

에러 없이
업로드 되는지 확인

```
bash
```

[Copy code](#)

```
docker push gyeongnamit202411/movie_recommend_frontend
```

이 명령어는 내 컴퓨터에 있는 도커 이미지(프로그램 실행 환경)를 인터넷의 저장소(Docker Hub)에 업로드(저장)하는 명령어입니다.

쉽게 말해, 내가 만든 요리(프로그램 환경)를 클라우드 냉장고(인터넷 저장소)에 보관하는 과정입니다.

이렇게 하면 다른 사람이나 다른 컴퓨터에서도 이 요리를 가져다 쓸 수 있습니다.


```
bash
```

[Copy code](#)

```
docker push gyeongnamit202411/movie_recommend_frontend
```

명령어 구성 살펴보기

- **docker push**

- 도커 이미지(완성된 프로그램)를 인터넷에 업로드하는 명령어입니다.
- 예: 햄버거를 만들어서 인터넷 냉장고(Docker Hub)에 저장합니다.

- **yeongnamit202411/movie_recommend_frontend**

- 업로드할 도커 이미지의 이름입니다.
- yyeongnamit202411 → 도커 허브 아이디(사용자 이름).
- movie_recommend_frontend → 프로그램 이름.예:

- 'yeongnamit202411'이라는 요리사(사용자)가 만든 'movie_recommend_frontend'(요리 메뉴)를 인터넷에 올립니다.

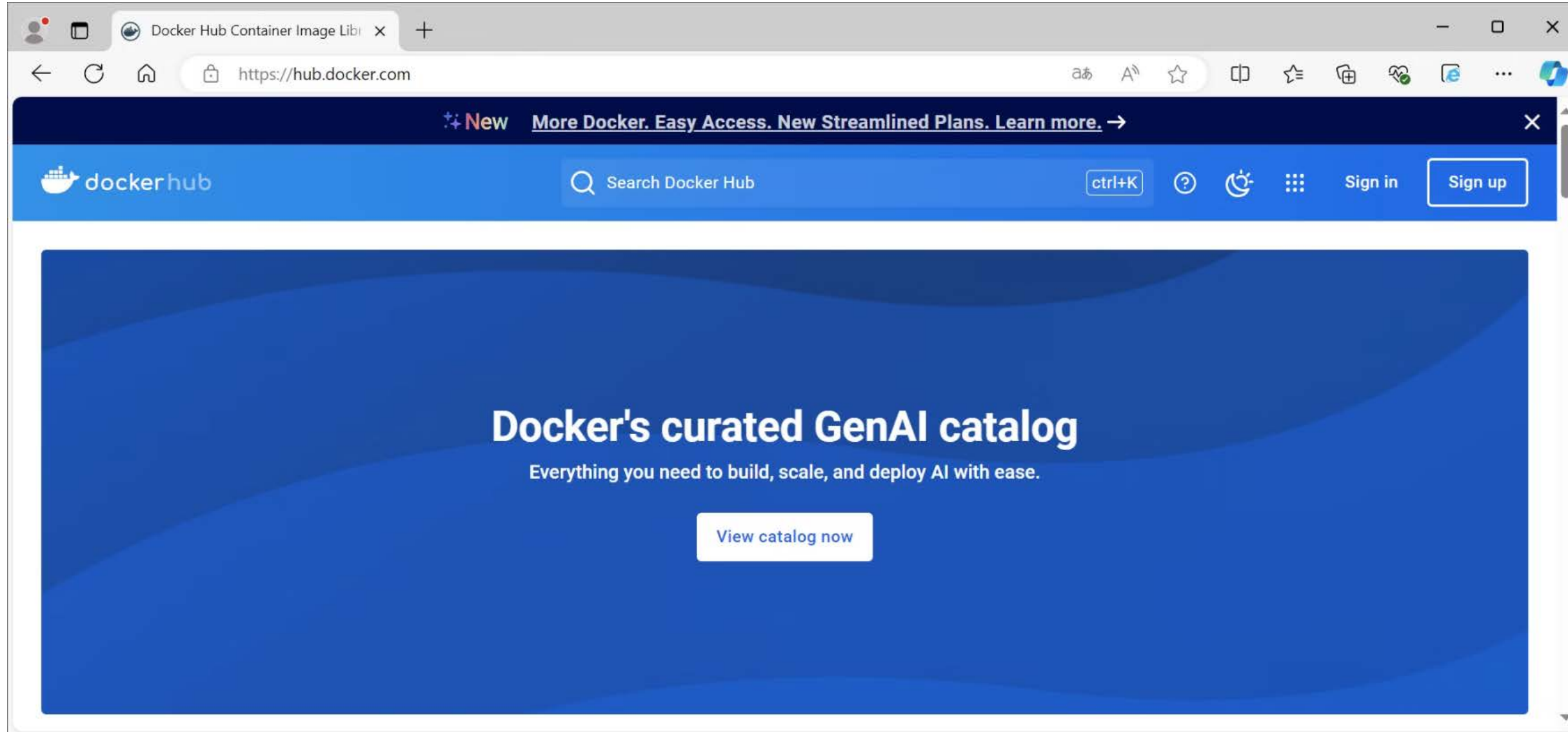
3.DockerHub 이미지 등록 확인

Docker 이미지가 생성되어 Docker Hub 저장소에 업로드됩니다. 이는 CI/CD 파이프라인의 중요한 단계입니다.

Docker Hub에 접속하여 새로운 이미지가 성공적으로 업로드되었는지 확인하고, 이미지 태그와 버전이 올바르게 지정되었는지 검증합니다. 이를 통해 애플리케이션 배포를 위한 준비가 완료되었음을 확인할 수 있습니다.

DockerHub 이미지 등록 확인

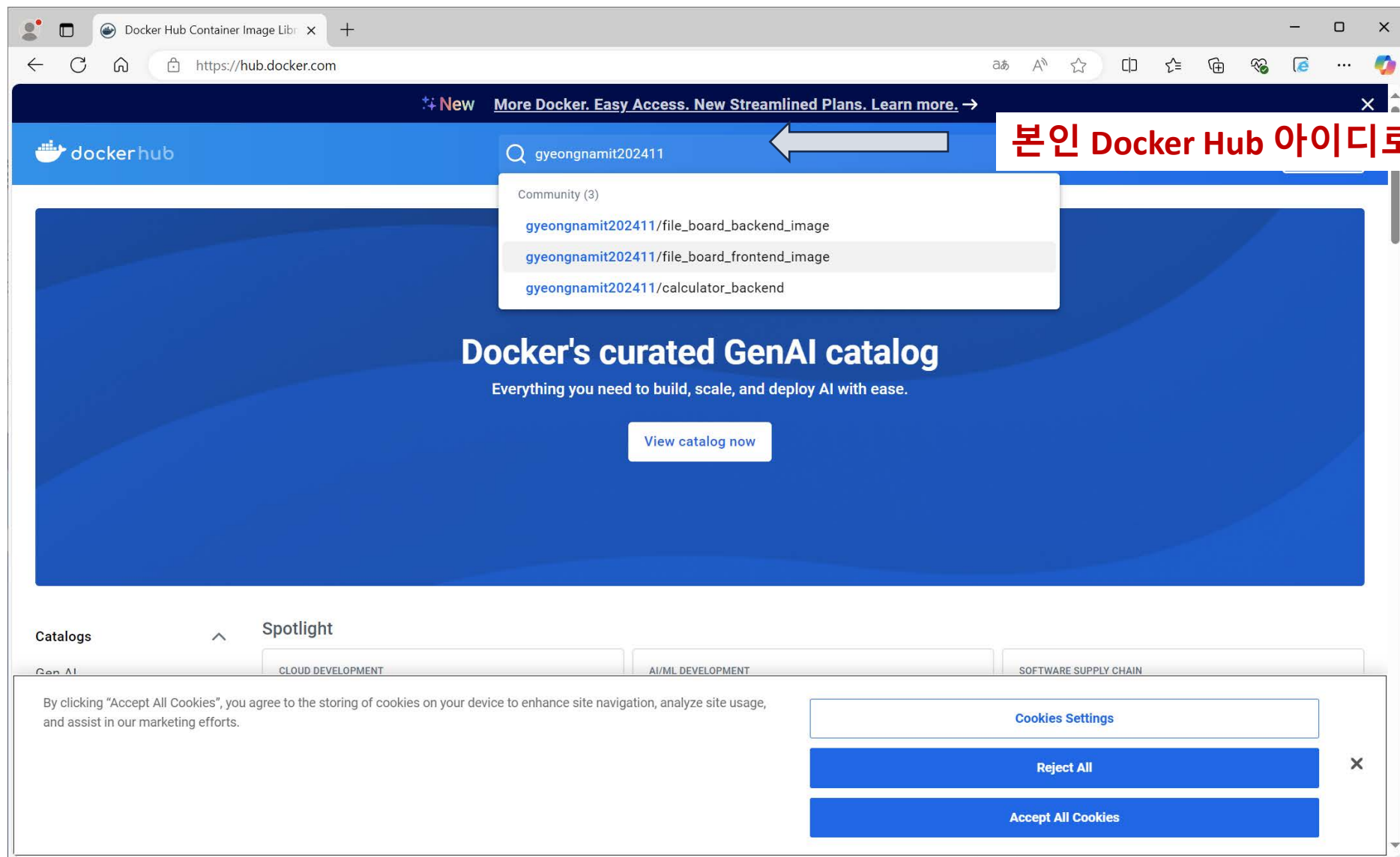
<https://hub.docker.com/> 접속 합니다



DockerHub 이미지 등록 확인

본인 Docker Hub 아이디로 검색 합니다

gyeongnamit202411 가 아이디 라면 gyeongnamit202411 로 검색 합니다



이미지 생성 확인

본인 Docker Hub 아이디/movie_recommend_frontend 가 검색 되는지 확인 합니다

The screenshot shows a web browser window with the Docker Hub search page. The search query is 'gyeongnamit202411'. The results show three items:

- gyeongnamit202411/movie_recommend_frontend** (By gyeongnamit202411 • Updated a minute ago) - This item is highlighted with a red arrow and a red text label.
- gyeongnamit202411/movie_recommend_backend** (By gyeongnamit202411 • Updated 10 hours ago)
- gyeongnamit202411/movie_recommend_frontend_step1** (By gyeongnamit202411 • Updated 12 hours ago)

The left sidebar shows filters for Products (Images, Extensions, Plugins), Trusted content (Docker Official Image, Verified Publisher, Sponsored OSS), and Categories (API Management, Content Management System).

본인 Docker Hub 아이디/movie_recommend_frontend 확인

이미지 생성 확인

본인 Docker Hub 아이디/movie_recommend_frontend 클릭 합니다

The screenshot shows a web browser window with the Docker Hub search results for the user 'gyeongnamit202411'. The search results are sorted by 'Recently updated' and show 7 results. The first result, 'gyeongnamit202411/movie_recommend_frontend', is highlighted with a red box. The second result is 'gyeongnamit202411/movie_recommend_backend' and the third is 'gyeongnamit202411/movie_recommend_frontend_step1'. The left sidebar shows filters for Products (Images, Extensions, Plugins), Trusted content (Official Image, Verified Publisher, Sponsored OSS), and Categories (API Management, Content Management System).

Explore Docker's Container Image x +

https://hub.docker.com/search?q=gyeongnamit202411&sort=updated_at&order=desc

New More Docker. Easy Access. New Streamlined Plans. Learn more. →

dockerhub

gyeongnamit202411

Sign in Sign up

Filters

1 - 7 of 7 results for gyeongnamit202411.

Recently updated

Products

- ☐ Images
- ☐ Extensions
- ☐ Plugins

Trusted content

- ☐ Docker Official Image ⓘ
- ☐ Verified Publisher ⓘ
- ☐ Sponsored OSS ⓘ

Categories

- ☐ API Management
- ☐ Content Management System

gyeongnamit202411/movie_recommend_frontend

By gyeongnamit202411 • Updated a minute ago

↓5 • ☆0

gyeongnamit202411/movie_recommend_backend

By gyeongnamit202411 • Updated 10 hours ago

↓34 • ☆0

gyeongnamit202411/movie_recommend_frontend_step1

By gyeongnamit202411 • Updated 12 hours ago

↓0 • ☆0

이미지 생성 확인

Docker 이미지가 생성 되었는지 확인 합니다

gyeongnamit202411/movie_recor

+

←

↺

🏠

🔒 https://hub.docker.com/r/gyeongnamit202411/movie_recommend_frontend

🔍

🌐

☆

📄

🔖

🔒

🔗

📧


⋮

🌈

New

More Docker. Easy Access. New Streamlined Plans. Learn more. →

×

 dockerhub

🔍

gyeongnamit202411

×

?

🔄

⋮


Sign in

Sign up

Explore

/

gyeongnamit202411/movie_recommend_frontend



gyeongnamit202411/movie_recommend_frontend


By [gyeongnamit202411](#) • Updated 2 minutes ago

IMAGE

☆0 ↓5

Overview

Tags



No overview available

This repository doesn't have an overview

Docker Pull Command

docker pull gyeongnamit202411/movie_recomm

commend_frontend

Copy