

# 로컬 LLM 완전 정복 가이드

## 실무자를 위한 단계별 설치 및 활용 매뉴얼

### 머리말

이 매뉴얼은 2025년 9월 현재, 비전문가도 스스로 AI 모델을 설치하고 활용할 수 있도록 제작된 실무 가이드입니다. 복잡한 용어와 기술을 배제하고, 실제 인턴십 과정에서 겪었던 시행착오와 문제 해결 과정을 상세하게 담았습니다. 이 문서를 통해 누구나 인터넷 연결 없이 안전하게 개인의 민감 정보를 다루는 로컬 AI 환경을 구축하고, 문서 기반 질의응답 (RAG) 기능을 실무에 즉시 적용할 수 있게 될 것입니다.

### 목차

- 로컬 AI 환경 구축 및 실행
  - 시스템 요구사항 및 준비 사항
  - Ollama 설치 및 기본 설정
  - Open WebUI 설치 및 접속
  - 설치 오류 및 문제 해결
- 문서 기반 질의응답 (RAG) 기능 심화 탐구
  - RAG 개념 및 필요성
  - 단일 문서 활용 RAG 테스트
  - 다수 문서 활용 RAG 테스트
  - 모델의 언어적 한계 분석 및 보고
- 모델 관리 및 최적화
  - LLM 모델 다운로드 및 연동
  - 모델별 특성 및 활용법
  - 성능 최적화 (메모리, 속도)
  - 업데이트 및 유지보수
- 부록
  - 용어 설명 (LLM, Docker, 컨테이너 등)
  - 명령어 모음집
  - 권장 하드웨어 사양 가이드

# 1. 로컬 AI 환경 구축 및 실행

## 1.1. 시스템 요구사항 및 준비사항

**개요:** 이 매뉴얼은 2025년 9월 현재, Windows 운영체제를 기준으로 작성되었으며, 다른 OS(Linux, macOS) 사용자는 각 OS에 맞는 설치 파일을 다운로드해야 합니다. 또한, 로컬 LLM 환경을 최초 구축하는 과정에서만 일시적인 인터넷 연결이 필요합니다. Ollama 모델 다운로드 및 Open WebUI 컨테이너 설치를 완료한 이후부터는 완전한 오프라인 상태에서 모든 기능을 사용할 수 있습니다. 더불어, 로컬 LLM 환경을 구축하기 위해 필요한 최소한의 시스템 사양과 준비 사항을 안내합니다.

- **운영체제:** Windows 11 (Ollama는 Windows 10 이상을 지원하지만, 11 버전을 권장합니다.)
- **메모리(RAM):** 최소 8GB, 권장 16GB 이상. 더 큰 모델을 구동하거나 여러 모델을 동시에 사용하려면 32GB 이상을 권장합니다.
- **저장 공간(SSD):** 최소 25GB 이상의 여유 공간이 필요합니다. 모델 크기에 따라 추가 공간이 필요할 수 있습니다.
- **그래픽 카드(선택):** GPU가 있으면 모델 추론 속도가 빨라지지만, 필수 사항은 아닙니다.

**준비사항:**

1. Ollama 설치 파일
2. Docker Desktop 설치 파일
3. 명령프롬프트(CMD) 또는 PowerShell

## 1.2. Ollama 설치 및 기본 설정

**과정:**

1. Ollama 공식 웹사이트(<https://ollama.com/>)에 접속하여 웹페이지 우측 상단의 Download 버튼을 눌러 해당 페이지로 접속한 후에, \*\*사용자의 OS에 알맞은 설치 파일(OllamaSetup.exe)\*\*을 다운로드합니다.  
(또는 다운로드 웹페이지(<https://ollama.com/download>)에 직접 접근하여 해당 설치 파일을 다운로드하도록 합니다.)

Models GitHub Discord Turbo

Search models

Sign in

Download



OpenAI and Ollama partner to launch gpt-4o

Chat & build with open models

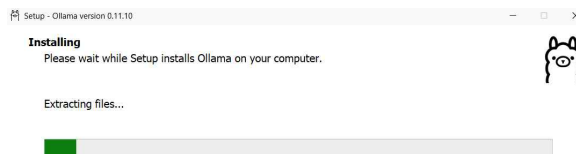
Download

Ollama 공식 웹사이트 접속 화면

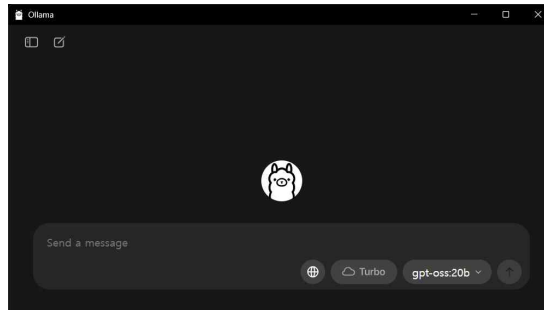


다운로드 완료된 설치 파일

2. 다운로드된 설치 파일을 실행하여 설치를 진행합니다.



3. 설치가 완료되면 Ollama 애플리케이션이 자동으로 실행됩니다.



Ollama 데스크톱 앱 실행 화면

#### 설치 확인:

1. 명령 프롬프트(CMD)를 열고 ollama를 입력한 후 Enter 키를 누릅니다.
2. Usage: ollama [command]와 함께 명령어 목록이 출력되면, Ollama가 올바르게 설치된 것입니다.

```

Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ >ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create      Create a model
  show        Show information for a model
  run         Run a model
  stop        Stop a running model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version    Show version information

Use "ollama [command] --help" for more information about a command.
  
```

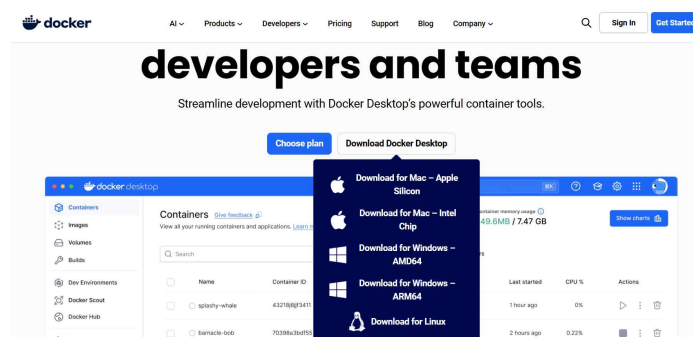
명령 프롬프트에서 ollama 명령어 실행 결과

### 1.3. Open WebUI 설치 및 접속

#### □ Docker Desktop 설치

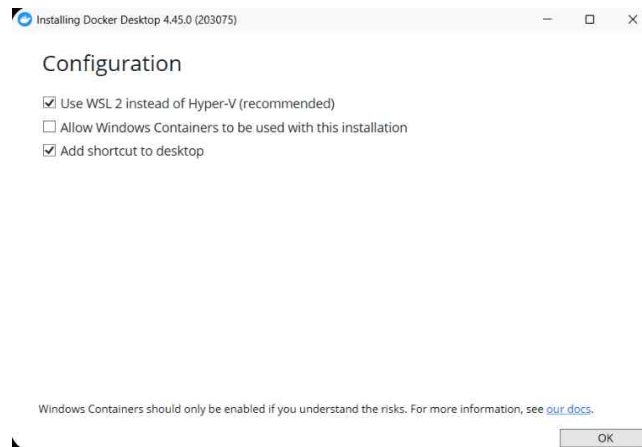
##### 과정:

1. Docker 공식 웹사이트(<https://www.docker.com/products/docker-desktop/>)에 접속하여 사용자 컴퓨터의 운영 체제에 맞는 파일을 다운로드합니다.



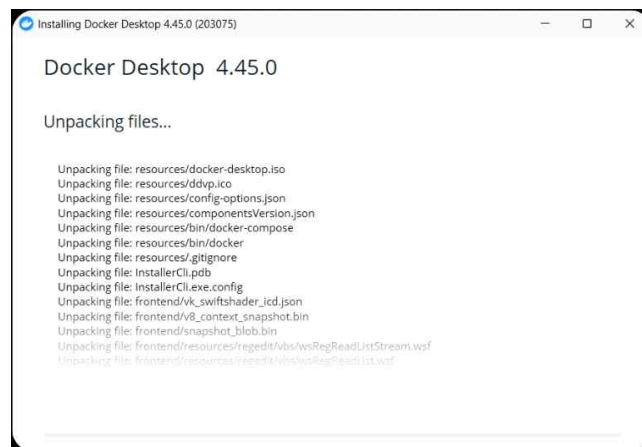
Docker Desktop 다운로드 페이지 화면

2. 다운로드한 설치 파일을 실행하여 설치를 진행합니다. 이 때, 기본 설정인 Use WSL 2 옵션을 그대로 두고 진행합니다.



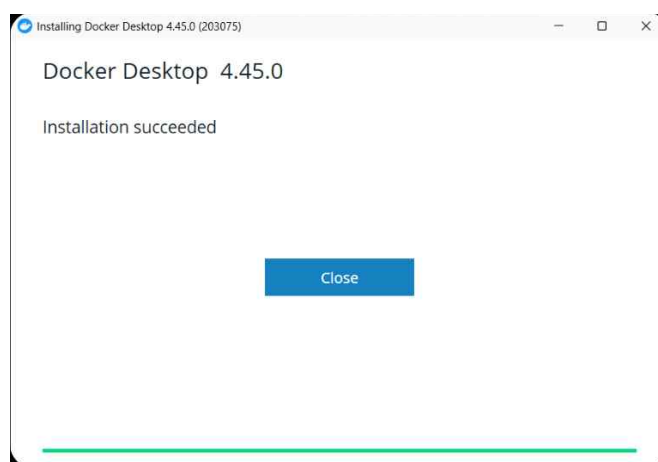
Docker Desktop 설치 Configuration 화면

3. 설치 과정이 진행되는 화면입니다.



Docker Desktop 설치 진행 중 화면

4. 설치가 완료되면 Installation succeeded 화면이 나타납니다.



Docker Desktop 설치 완료 화면

## □ Open WebUI 컨테이너 설치 및 실행

과정:

### 1. Docker Desktop 실행

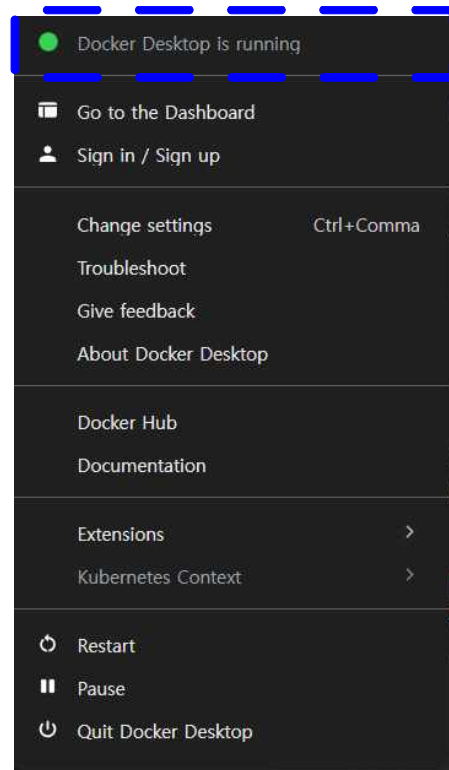
- Docker Desktop 애플리케이션을 실행하고, 모니터 화면 우측 하단 시스템 트레이 부분에 '고래 모양' 아이콘

에 마우스 우측 클릭하여 임시 메뉴가 뜨면 가장 위에 Docker Desktop is running 메시지가 뜨는지 확인합니다.

- 아래 이미지를 참고하여 정상적으로 실행되었는지 확인하십시오.



Docker Desktop 애플리케이션(시스템 트레이)



마우스 우클릭 후 임시 메뉴를 통한 'Docker Desktop is running' 메시지 확인

## 2. 명령 프롬프트(CMD) 실행

- Windows 11 사용자는 모니터 화면 하단의 시작 버튼 옆의 돋보기 아이콘을 클릭합니다. 또는 키보드 단축키 Windows + S를 누른 후, 검색창에 CMD 또는 명령 프롬프트를 입력하여 실행합니다.
- 이제 이 창에 아래 명령어들을 순서대로 입력하여 Open WebUI 환경을 구축합니다.

## 3. Ollama 네트워크 생성

- 명령 프롬프트에서 아래 명령어를 입력하여 ollama 네트워크를 생성합니다.

(Tip: 기존에 네트워크가 생성되었으면 Error response from daemon 메시지가 뜨지만 무시해도 됩니다.)

```
docker network create ollama
```

## 4. Open WebUI 컨테이너 설치 및 실행

- 아래 명령어를 입력하여 Open WebUI를 설치 및 실행합니다.

(Tip: 이 명령어는 Open WebUI 컨테이너를 설치하고, Ollama와 통신할 수 있도록 연결하며, PC 재부팅 시에도 자동으로 실행되도록 설정합니다.)

```
docker run -d --network=ollama --pull=always -v open-webui:/app/backend/data -p 8080:8080 --name open-webui --restart unless-stopped ghcr.io/open-webui/open-webui:main
```

### ※ 실습 내용:

아래 스크린샷 이미지는 실습 당시 docker run -d -p 3000:8080 --add-host... 명령어로 Open WebUI를 설치하고 실행했던 화면입니다.

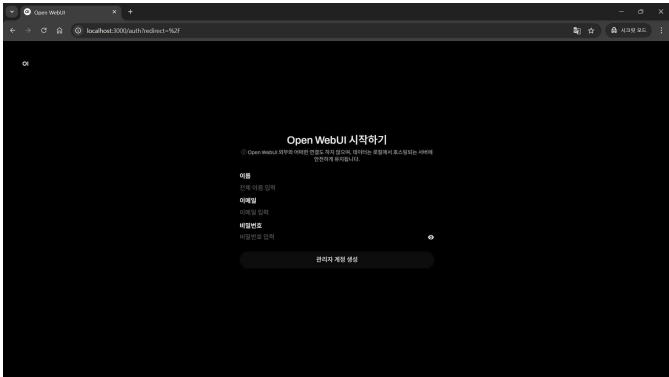
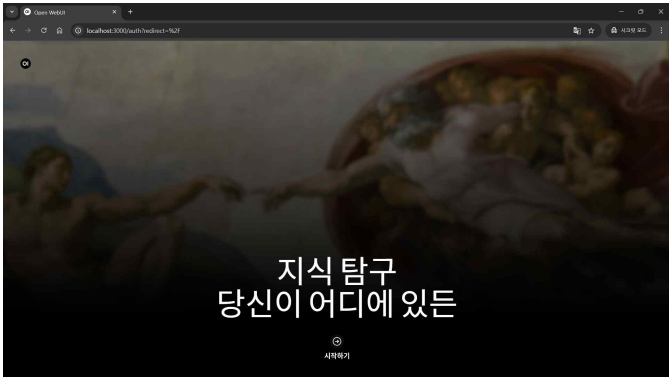
이는 컨테이너가 외부 네트워크를 통해 로컬 PC의 Ollama와 통신하는 방식이라 효율성과 안정성이 떨어집니다. 따라서 최종 매뉴얼에는 Docker의 권장 사항인 \*\*docker network\*\*를 활용하는 명령어를 기재했습니다. 이 방식으로

진행하면 더 견고한 로컬 LLM 환경을 구축할 수 있습니다.

```
관리자: 명령 프롬프트
C:\Windows\System32>docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/a
pp/backend/data --name open-webui --restart unless-stopped ghcr.io/open-webui/open-webui:main
Unable to find image 'ghcr.io/open-webui/open-webui:main' locally
main: Pulling from open-webui/open-webui
a1c0fae3cd32: Pull complete
f8f3a467bf93: Pull complete
b32275e9f783: Pull complete
4f4fb700ef54: Pull complete
227ace10ab92: Pull complete
3812b12828ba: Pull complete
7805c959c264: Pull complete
a647374e6be3: Pull complete
93891a47222a: Pull complete
8c6b15711d8b: Pull complete
d107e437f729: Pull complete
d5c111aa61fa: Pull complete
659036234d55: Pull complete
1ce95d9ae66c: Pull complete
ce4392d98604: Pull complete
Digest: sha256:2e78a2f9f6f62173ae28d2203f3c9bcdadc614023380ebbe903ea9fab772535e
Status: Downloaded newer image for ghcr.io/open-webui/open-webui:main
bde46e474d899aeea2001e81ef0781e9e3e332a8bf34b44b7c5588211c735fa
```

명령 프롬프트에서 docker run 명령어 실행 결과

5. 웹 브라우저 접속: 웹 브라우저를 열고 주소창에 localhost:8080을 입력하여 Open WebUI에 접속합니다.
- 정상적으로 계정 생성 페이지가 나타나는 것을 확인할 수 있으며, 입력란에 내용을 입력한 뒤 계정 생성을 완료합니다.



웹 브라우저에서 Open WebUI 계정 생성 페이지 접속 화면

◇ 설치 오류 및 문제 해결

1. network ollama not found 오류

- **오류 원인:** docker run 명령어를 실행하기 전에 docker network create ollama 명령어를 실행하지 않았기 때문입니다.
- **해결 방법:** docker network create ollama 명령어를 다시 실행하여 네트워크를 생성한 후, Open WebUI 설치 명령어를 재실행하면 문제가 해결됩니다.

2. localhost:8080 접속 오류

- **오류 원인:** Docker Desktop 애플리케이션이 정상적으로 실행 중이지 않기 때문입니다.
- **해결 방법:** 모니터 화면 우측 하단 시스템 트레이의 고래 모양 아이콘을 우클릭하여 Docker Desktop is running 메시지가 뜨는지 확인합니다.

## 2. 문서 기반 질의응답 (RAG) 기능 심화 탐구

### 2.1. RAG 개념 및 필요성

개요: RAG는 '검색 증강 생성' 기술로, LLM이 외부 문서(PDF, 웹페이지 등)에서 정보를 **검색(Retrieval)**하여 답변을 **생성(Generation)**하는 방식입니다. 이를 통해 모델의 기본 지식이 아닌, 특정 문서의 내용을 바탕으로 정확한 답변을 얻을 수 있습니다.

### 2.2. 단일 문서 활용 RAG 테스트

과정:

Open WebUI 대화창에서 우측 상단의 종이 아이콘을 클릭합니다.

'파일 업로드' 버튼을 선택하여 가지고 있는 PDF 파일을 업로드합니다.

업로드된 문서를 llama2 모델이 자동으로 학습합니다.

실습 내용:

Open WebUI에 PDF 문서를 업로드하는 화면

ROS2\_Open\_Seminar\_교재.pdf 파일을 업로드하고 **"ROS2의 특징은 무엇인가요?"**라고 질문하여 답변을 받는 것을 확인했습니다.

### 2.3. 다수 문서 활용 RAG 테스트

개요: 하나의 PDF 문서만 활용했던 실습에서 나아가, 여러 개의 문서를 동시에 학습시켜 로컬 LLM의 RAG 기능이 다양한 지식을 통합하고 활용할 수 있는지 테스트했습니다.

과정:

테스트용 PDF 파일 준비: Ollama 모델의 다중 문서 처리 능력을 검증하기 위해 서로 다른 주제의 PDF 파일 5개를 준비했습니다.

반려동물 건강 관리 가이드

인공지능 기초 용어 설명

로마 시대 건축물 소개

커피 원두의 종류와 특징

2025년 주요 IT 트렌드

PDF 동시 업로드: Open WebUI 대화창에 위 5개의 PDF 파일을 동시에 업로드했습니다.

다중 문서 기반 질의응답: 업로드된 모든 문서의 내용을 포괄하는 질문을 한국어로 입력하여 답변을 확인했습니다.

입력 질문: "5개 파일의 내용들을 융합하여 간략히 설명하고 중요한 핵심만 추려서 설명해줘"

모델 답변: 모델은 한국어 질문에도 불구하고 영어로 답변을 생성했습니다. 하지만 답변 내용은 5개 PDF 파일의 핵심 내용을 모두 정확하게 요약하고 추출했습니다. 이는 llama2 모델이 언어적 한계에도 불구하고 RAG 기능 자체는 완벽하게 수행했음을 증명합니다.