

Chapter 1. Langflow Hands on

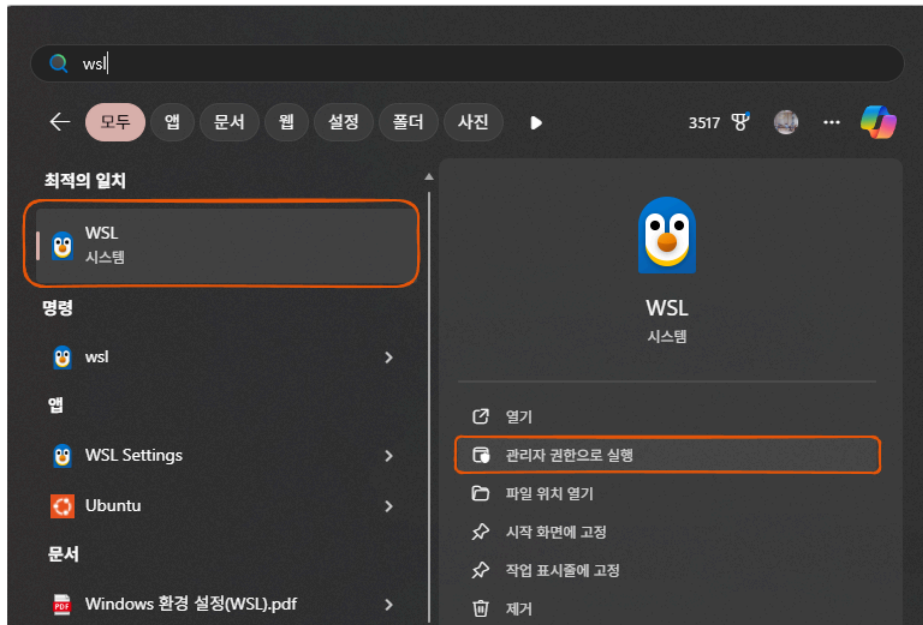
1. 개발 환경 구성 (Windows + WSL2 + Ubuntu 24.04)

오늘 실습은 Windows 10/11에서 WSL2를 활용해 Ubuntu 24.04 환경을 구성하고, 그 위에서 uv 로 Langflow OSS(1.6.8)를 실행하는 방식입니다.

Windows에 설치된 Python/Anaconda 환경과 독립적으로 작동하기 때문에 깔끔하게 시작할 수 있습니다.

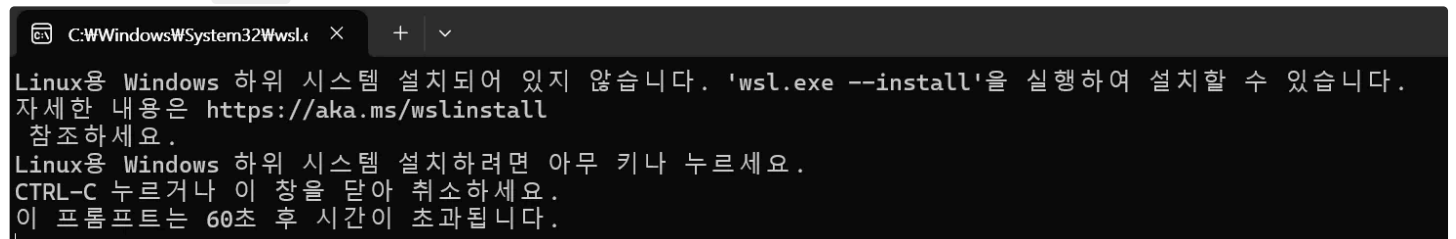
1-1. WSL2 설치하기

0. 윈도우에서 wsl 시작하기

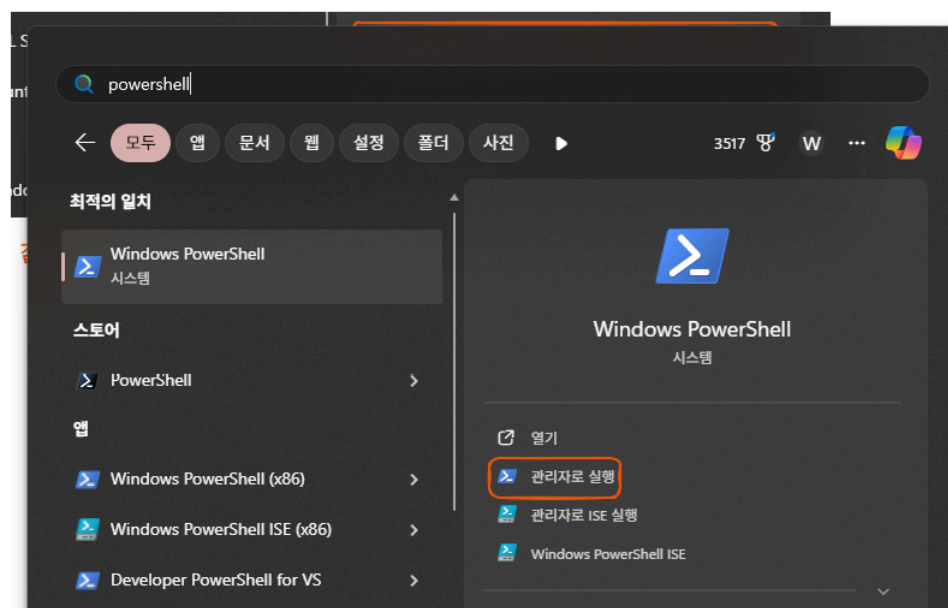


win 10도 같은 흐름

만약 아래 메시지 발생 시 아무 키나 눌러서 하위시스템 설치 후 재부팅 진행해주세요.



1. PowerShell을 관리자 권한으로 실행해주세요.



2. 아래 명령어를 입력해 WSL2와 Ubuntu 24.04를 설치합니다.

```
wsl --set-default-version 2
wsl --install -d Ubuntu-24.04
```

3. 설치가 완료되면 사용자와 비밀번호를 설정하라는 메시지가 나옵니다.

```
PS C:\WINDOWS\system32> wsl --set-default-version 2
WSL 2와의 주요 차이점에 대한 자세한 내용은 https://aka.ms/wsl2를 참조하세요

작업을 완료했습니다.
PS C:\WINDOWS\system32> wsl --install -d Ubuntu-24.04
다운로드 중: Ubuntu 24.04 LTS
[===== 19.0% ]
```

설치 완료 후, 자동 실행되지 않으면 직접 실행

```
wsl -d Ubuntu-24.04
```

```
PS C:\WINDOWS\system32> wsl -d Ubuntu-24.04
Provisioning the new WSL instance Ubuntu-24.04
This might take a while...
Create a default Unix user account: sdsclass
```

계정정보(자율) : sdsclass

비밀번호: 1234

✓ **wsl --install** 이후 최초 부팅 시 설정한 **Ubuntu 계정/비밀번호**를 반드시 기록해 두세요.

아래까지 하면 완료

```
new password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Nov 23 12:37:13 KST 2025

System load:  0.1          Processes:      29
Usage of /:   0.1% of 1006.85GB  Users logged in: 0
Memory usage: 3%          IPv4 address for eth0: 172.20.2.164
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/sdsclass/.hushlogin file.
sdsclass@woongs-root:/mnt/c/WINDOWS/system32$
```

4. 필요 시 재부팅을 진행합니다.

1-2. Ubuntu 초기화 및 필수 도구 설치

1. 시작 메뉴에서 "WSL" 또는 "Ubuntu 24.04"를 검색해 실행합니다.
2. (선택) 패키지 다운로드 속도 개선 - **Kakao Mirror** 설정
네트워크 환경이 느린 경우, 한국 내 미러 서버로 변경하면 설치 속도가 크게 향상됩니다.

기존 설정 백업

```
sudo cp /etc/apt/sources.list.d/ubuntu.sources /etc/apt/sources.list.d/ubuntu.sources.backup
```

Kakao Mirror로 변경 (archive.ubuntu.com → mirror.kakao.com)

```
sudo sed -i 's|http://archive.ubuntu.com|http://mirror.kakao.com|g' /etc/apt/sources.list.d/ubuntu.sources
```

⇒ 한국 환경에서만 미러 위치를 한국 CDN으로.

💡 이 단계는 선택사항입니다. 건너뛰어도 실습 진행에는 문제 없지만, 설치 시간이 다소 길어질 수 있습니다.

백업 복구(필요한 경우만)

```
sudo cp /etc/apt/sources.list.d/ubuntu.sources.backup /etc/apt/sources.list.d/ubuntu.sources
```

3. 터미널에서 아래 명령어로 최신 패키지과 개발 도구를 설치합니다.

```
sudo apt update && sudo apt full-upgrade -y
sudo apt install build-essential curl wget git -y
```

```
Setting up libstdc++12-dev:amd64 (12.1.0-6ubuntu2~24.04) ...
Setting up cpp-x86-64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up cpp-13 (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-13-x86-64-linux-gnu (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-13 (13.3.0-6ubuntu2~24.04) ...
Setting up cpp (4:13.2.0-7ubuntu1) ...
Setting up g++-13-x86-64-linux-gnu (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-x86-64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up gcc (4:13.2.0-7ubuntu1) ...
Setting up g++-x86-64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up g++-13 (13.3.0-6ubuntu2~24.04) ...
Setting up g++ (4:13.2.0-7ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.10ubuntu1) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.6) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4.1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
N: Ignoring file 'ubuntu.sources.backup' in directory '/etc/apt/sources.list.d/' as it has an invalid filename extension
sdsclass@woongs-root:~$ |
```

4. Python 패키지 관리 도구인 `uv` 를 설치한 후, 셸을 재시작합니다.

```
curl -LsSf https://astral.sh/uv/install.sh | sh
exec bash
uv --version
```

```
sdsclass@woongs-root:~$ curl -LsSf https://astral.sh/uv/install.sh | sh
exec bash
uv --version
downloading uv 0.9.11 x86_64-unknown-linux-gnu
no checksums to verify
installing to /home/sdsclass/.local/bin
uv
uvx
everything's installed!

To add $HOME/.local/bin to your PATH, either restart your shell or run:

source $HOME/.local/bin/env (sh, bash, zsh)
source $HOME/.local/bin/env.fish (fish)
sdsclass@woongs-root:~$ |
```

```
source $HOME/.local/bin/env
```

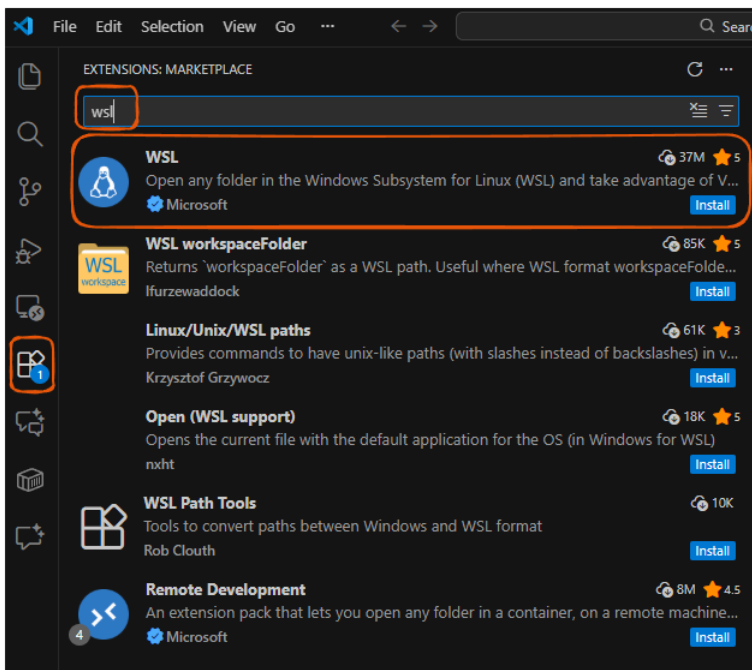
5. Langflow가 필요로 하는 Python 3.12를 미리 설치해 둡니다.

```
uv python install 3.12
```

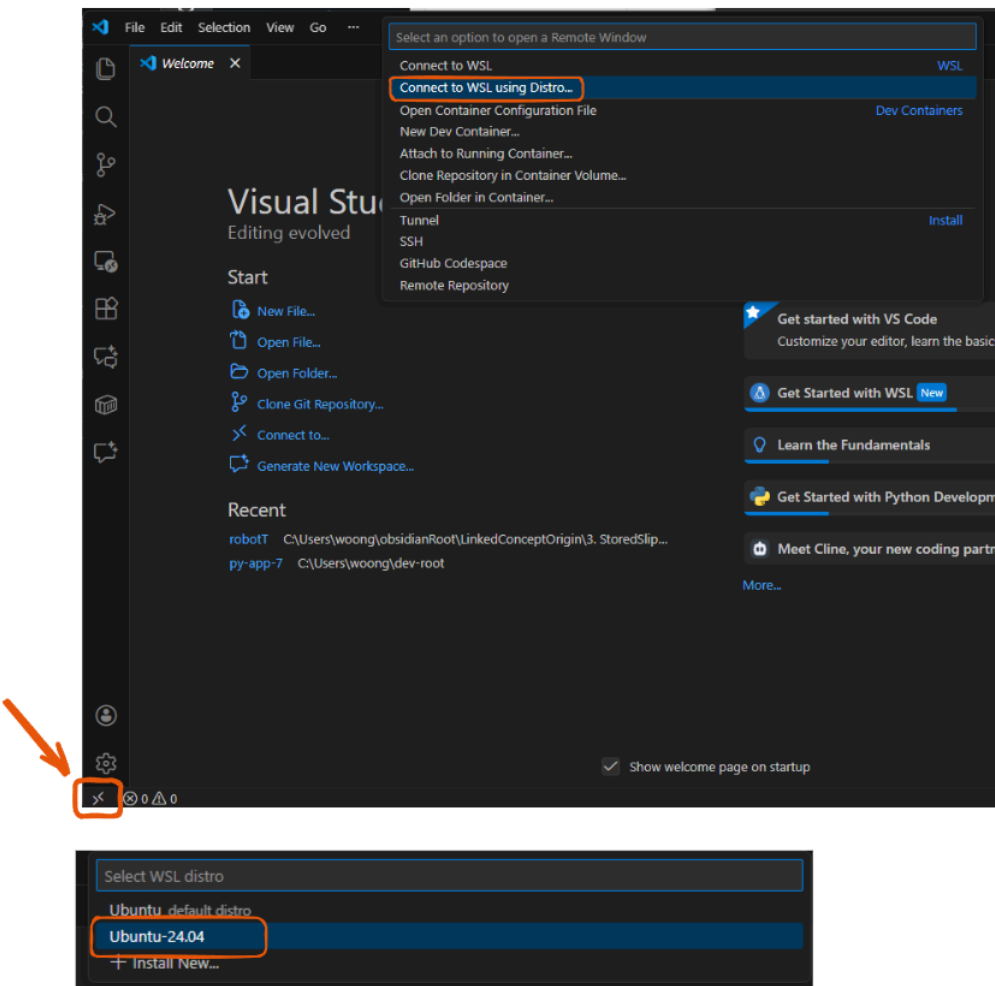
1-3. VS Code와 Git 연동

1. Windows에 VS Code를 설치한 후, WSL 확장(Remote - WSL) 을 추가합니다.

[Download Visual Studio Code - Mac, Linux, Windows](#)



2. VS Code 왼쪽 하단의 <WSL> 아이콘을 클릭한 뒤 "WSL: Ubuntu-24.04에 새 창 열기"를 선택합니다.



3. 터미널을 열고(단축키: `Ctrl+``) 실행 저장소를 클론합니다.

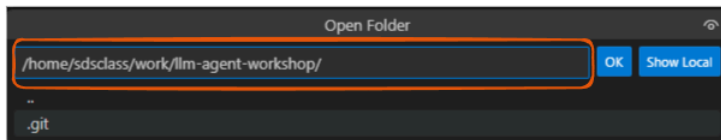
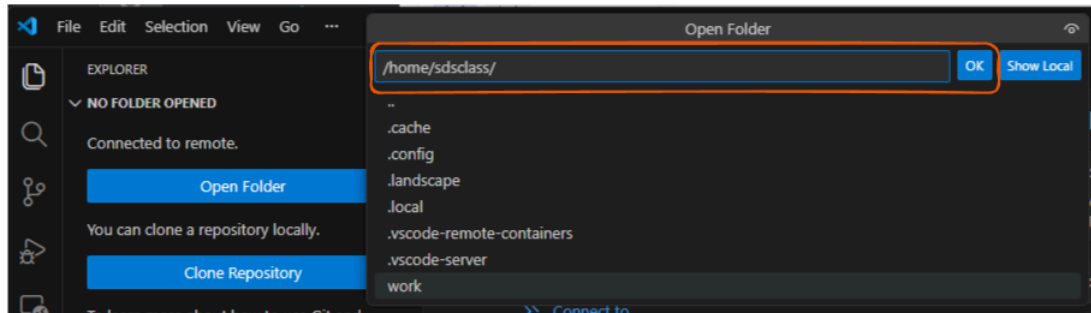
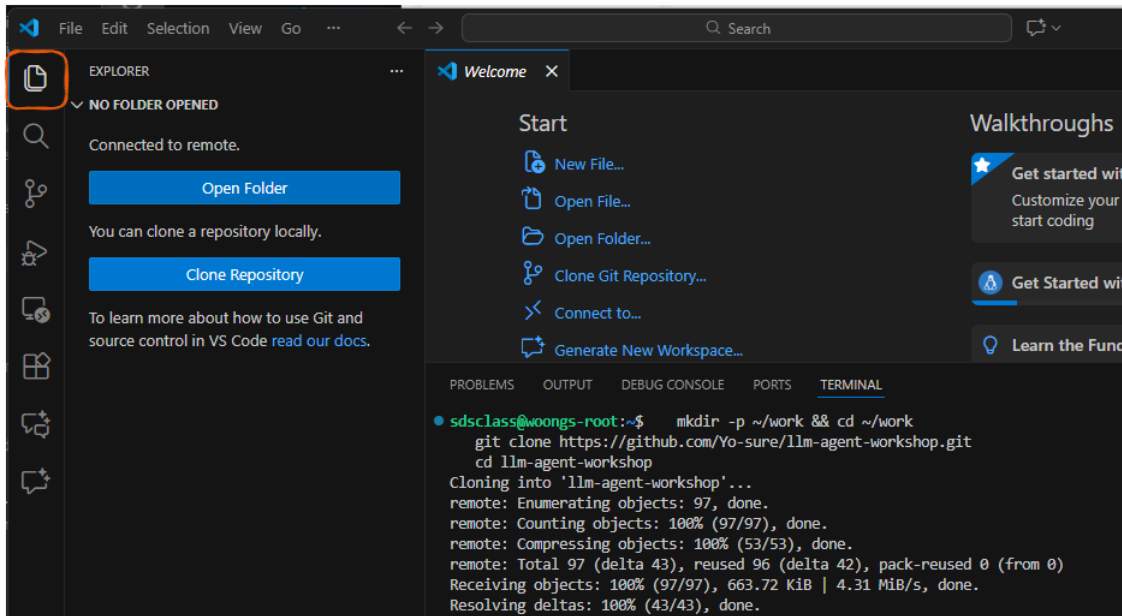
```
mkdir -p ~/work && cd ~/work
git clone https://github.com/Yo-sure/llm-agent-workshop.git
cd llm-agent-workshop
```

```

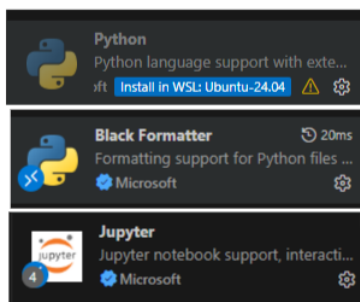
sdsclass@woongs-root:~$ mkdir -p ~/work && cd ~/work
git clone https://github.com/Yo-sure/llm-agent-workshop.git
cd llm-agent-workshop
Cloning into 'llm-agent-workshop'...
remote: Enumerating objects: 97, done.
remote: Counting objects: 100% (97/97), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 97 (delta 43), reused 96 (delta 42), pack-reused 0 (from 0)
Receiving objects: 100% (97/97), 663.72 KiB | 4.31 MiB/s, done.
Resolving deltas: 100% (43/43), done.
sdsclass@woongs-root:~/work/llm-agent-workshop$ ll
total 1364
drwxr-xr-x 3 sdsclass sdsclass 4096 Nov 23 13:25 ./
drwxr-xr-x 3 sdsclass sdsclass 4096 Nov 23 13:25 ../
drwxr-xr-x 8 sdsclass sdsclass 4096 Nov 23 13:25 .git/
-rw-r--r-- 1 sdsclass sdsclass 2706 Nov 23 13:25 .gitignore
-rw-r--r-- 1 sdsclass sdsclass 430 Nov 23 13:25 .replit
-rw-r--r-- 1 sdsclass sdsclass 1479 Nov 23 13:25 README.md
-rw-r--r-- 1 sdsclass sdsclass 9927 Nov 23 13:25 generated-icon.png
-rw-r--r-- 1 sdsclass sdsclass 0 Nov 23 13:25 main.py
-rw-r--r-- 1 sdsclass sdsclass 368 Nov 23 13:25 pyproject.toml
-rw-r--r-- 1 sdsclass sdsclass 1354973 Nov 23 13:25 uv.lock

```

4. 폴더에서 열어줍니다.



5. VS Code 확장 탭에서 Python, Jupyter, Black Formatter를 찾아 "Install on WSL" 버튼으로 설치합니다.



1-4. Langflow OSS 실행하기

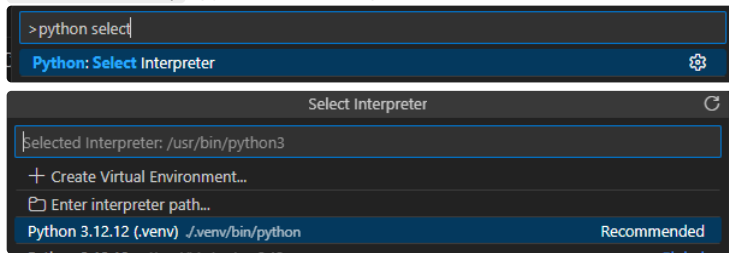
1. 프로젝트 루트에서 가상환경을 생성하고 의존성을 설치합니다.

```
uv venv .venv
uv sync
```

의존성 설치에는 몇 분이 소요될 수 있습니다

- 설치되는 동안 코드의 python 가상환경을 에디터에 묶어줍니다.

ctrl + shift + p : python select interpreter 검색

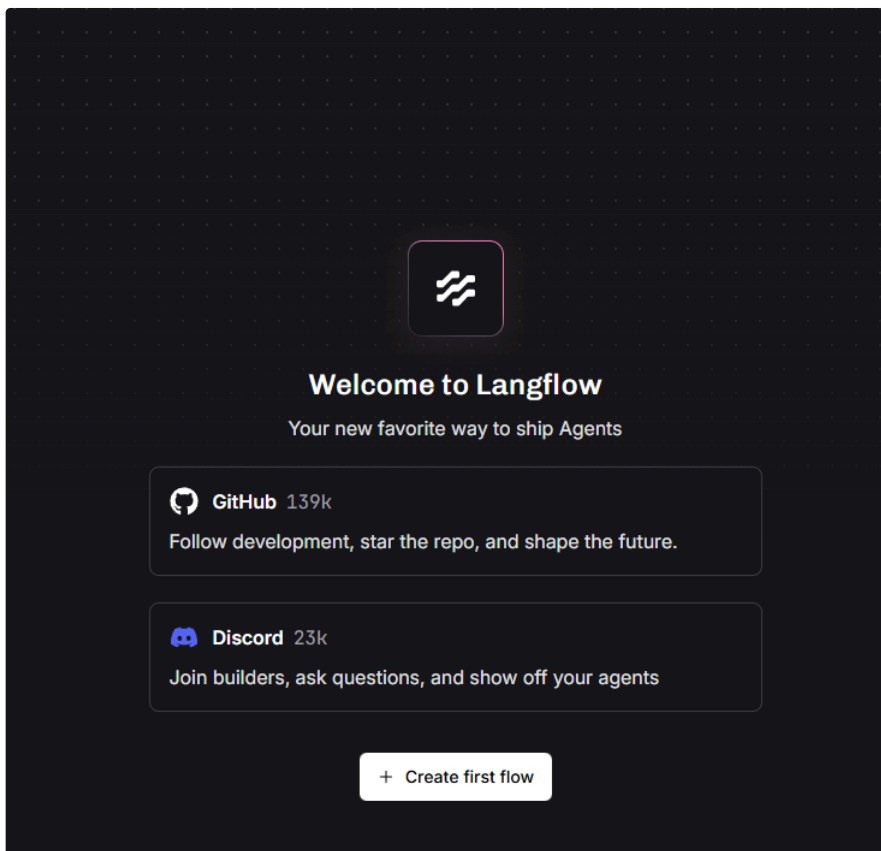
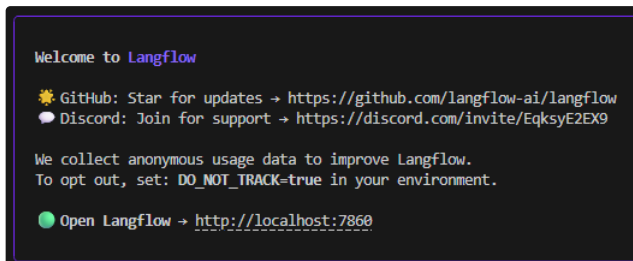


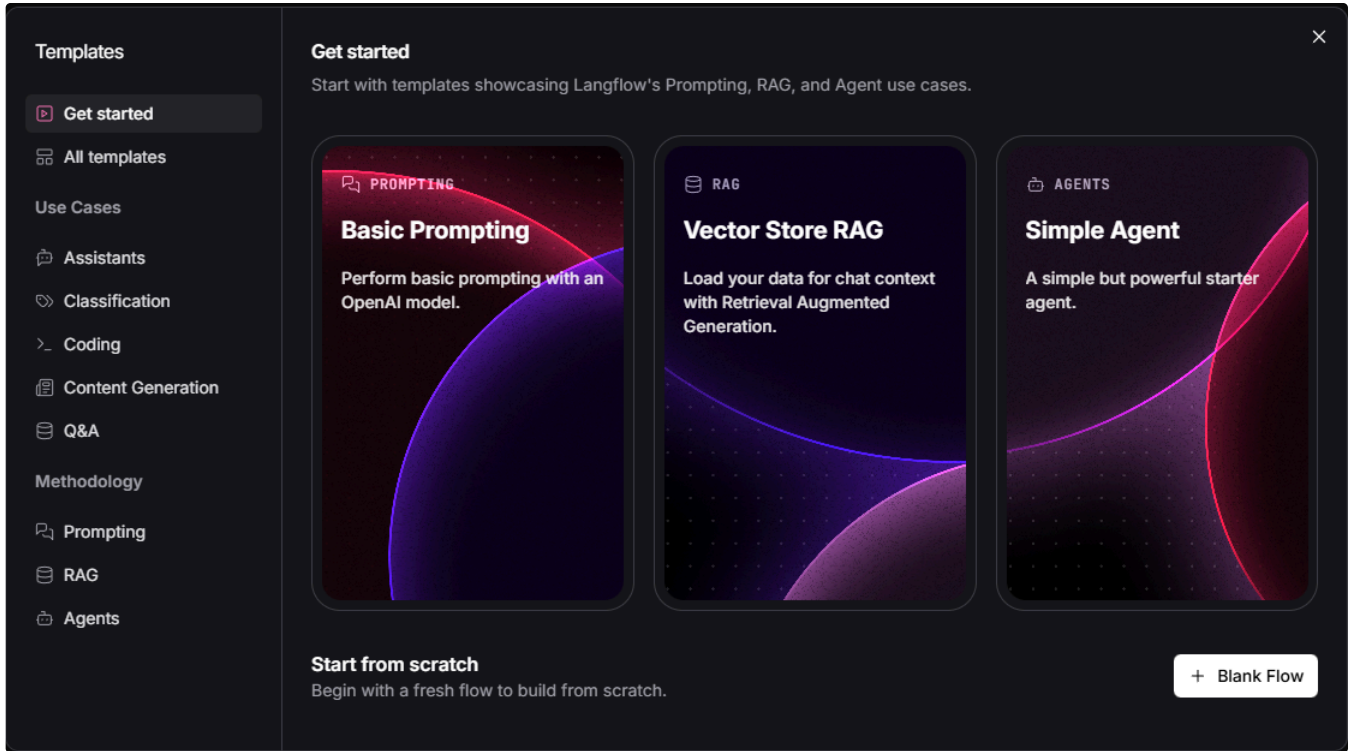
- Langflow 버전을 확인한 후 실행합니다.

```
uv run langflow --version # 1.6.8 출력 확인
```

```
uv run langflow run
```

- 터미널에 표시되는 URL(<http://127.0.0.1:7860>)을 브라우저에서 열면 Langflow UI가 나타납니다.





1-5. 브랜치 로드맵

실습은 Git 브랜치를 이동하며 진행됩니다. 각 단계에 맞는 코드와 예제가 준비되어 있습니다.

예시

단계	설명	명령
main	WSL 환경 세팅 및 UI 투어	<code>git checkout main</code>
01-news-agent	GDEL트 뉴스 분석 도구 실습	<code>git checkout 01-news-agent</code>

LLM API 키는 실습 중 제공되는 별도 링크(Notion)에서 확인하실 수 있습니다.

<https://bit.ly/251124AGENT>

✓ API 키는 `.env` 파일 또는 Langflow 환경 변수 입력란에만 보관해 주세요. Git에 절대 올리지 마세요.

🕒 Simple Agent 만들어보기

다음 단계

환경 설정이 완료되었습니다! 이제 **Chapter 2. Agent Flow & GDEL트** 문서로 이동해서 실습을 진행합니다.