



# 산업공학과 서버 사용 튜토리얼

2023/10/13

이 유 진

yjlee9040@yonsei.ac.kr

# 서버 사용 안내

---

- 현재 **구글폼**으로 서버 신청을 받고 있습니다. N주차에 대한 사용 신청은 n-1주차 일요일 자정까지 신청을 받고 있으며 당일 신청 및 해당 주에 대한 신청을 그 주에 받는 것은 불가하오니 참고 부탁드립니다.
- 산업공학과 서버 접속은 교내 ip에서만 가능하므로, 자택을 비롯한 **학교 이외의 공간에서는 사용이 제한됩니다.**
- 도커 컨테이너 환경에서 작업을 해야 원활한 서버 내 작업이 가능합니다. 참고 부탁드립니다.

# 실습 환경

---

Visual Studio Code



Visual Studio Code

Version: 1.76.1 (user setup)

Commit: 5e805b79fcb6ba4c2d23712967df89a089da575b

Date: 2023-03-08T16:32:00.131Z (2 wks ago)

Electron: 19.1.11

Chromium: 102.0.5005.196

Node.js: 16.14.2

V8: 10.2.154.26-electron.0

OS: Windows\_NT x64 10.0.19045

Sandboxed: Yes

Copy

OK

# 1. SSH 연결

# Contents

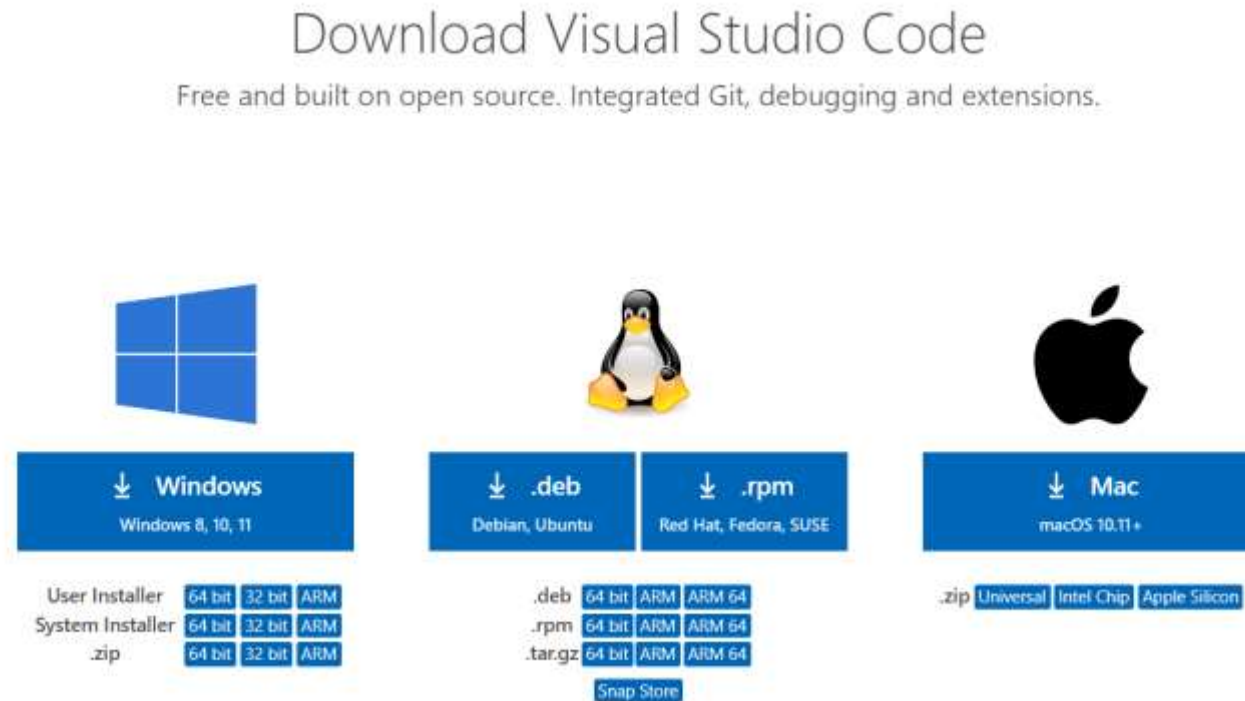
---

1. SSH 연결
2. 도커 컨테이너 생성
3. 도커 컨테이너 VSC에 연결
4. VSC에서 주피터 노트북 파일 생성

# SSH 연결

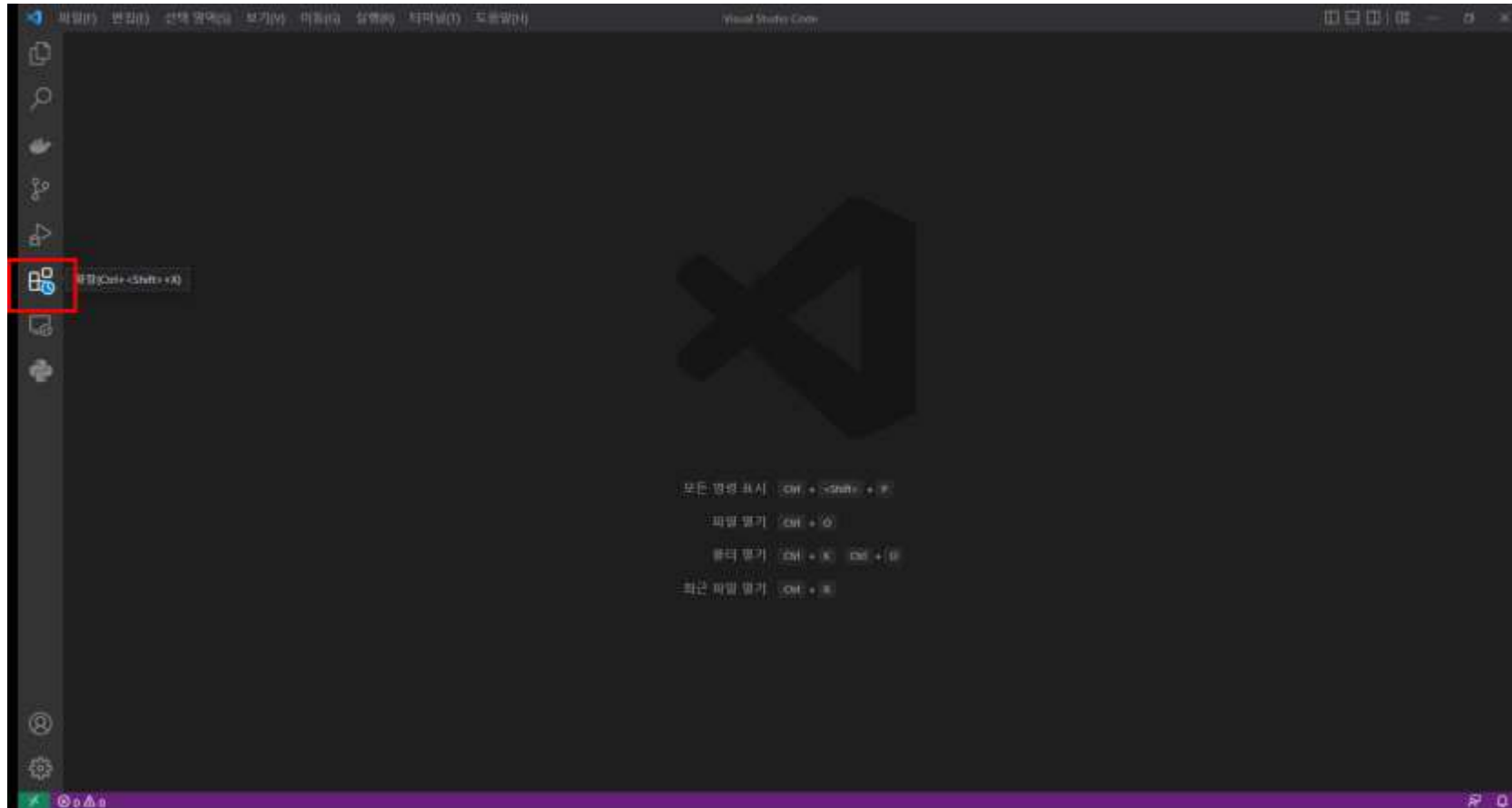
## 1. VSC 설치 (본인의 OS에 맞게)

<https://code.visualstudio.com/download>



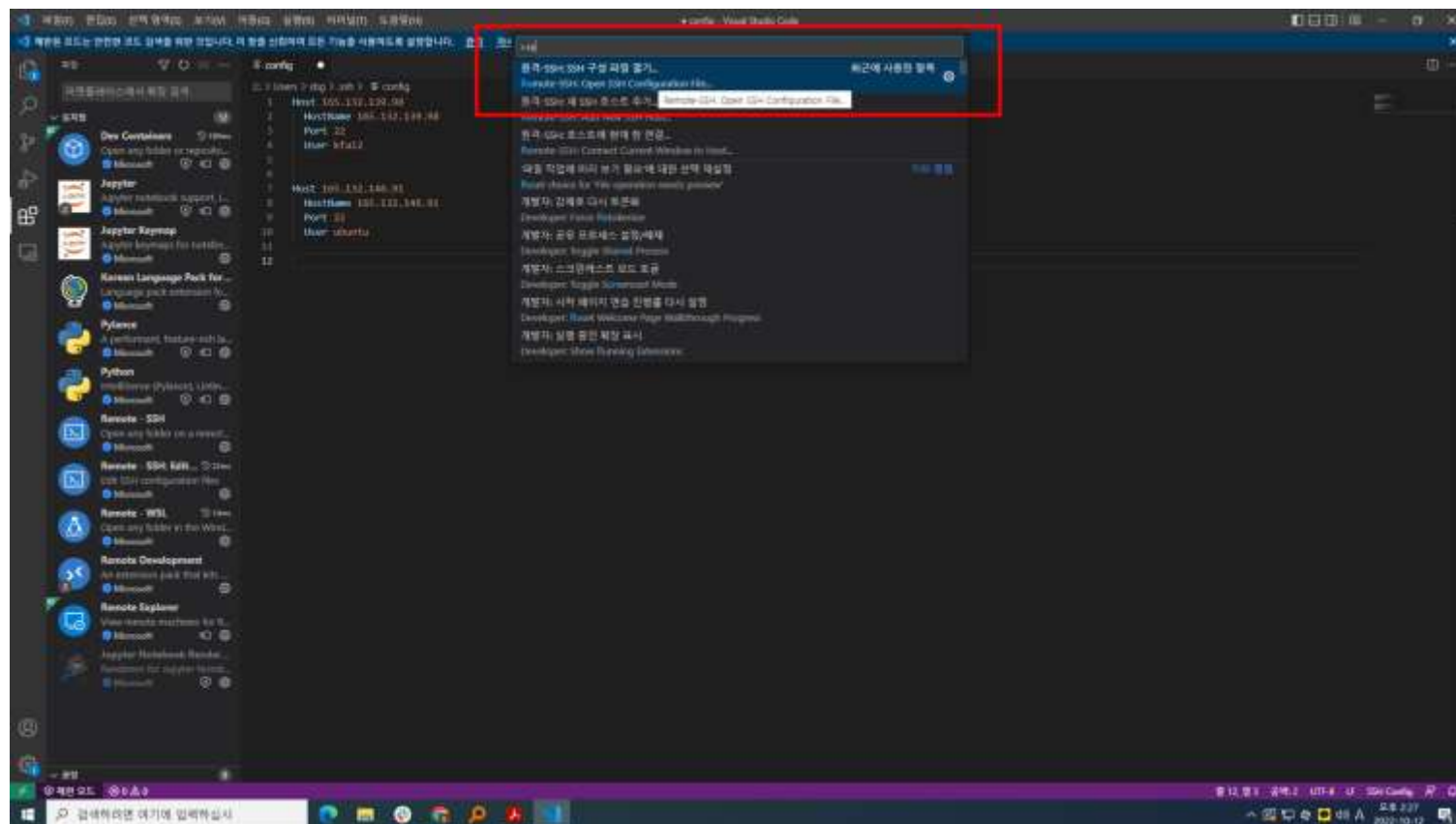
# SSH 연결

## 2. VSC 들어가서 extensions → remote-ssh 설치



# SSH 연결

3. Ctrl + shift + p (명령 팔레트) → remote-ssh open ssh configuration 클릭





# SSH 연결

---

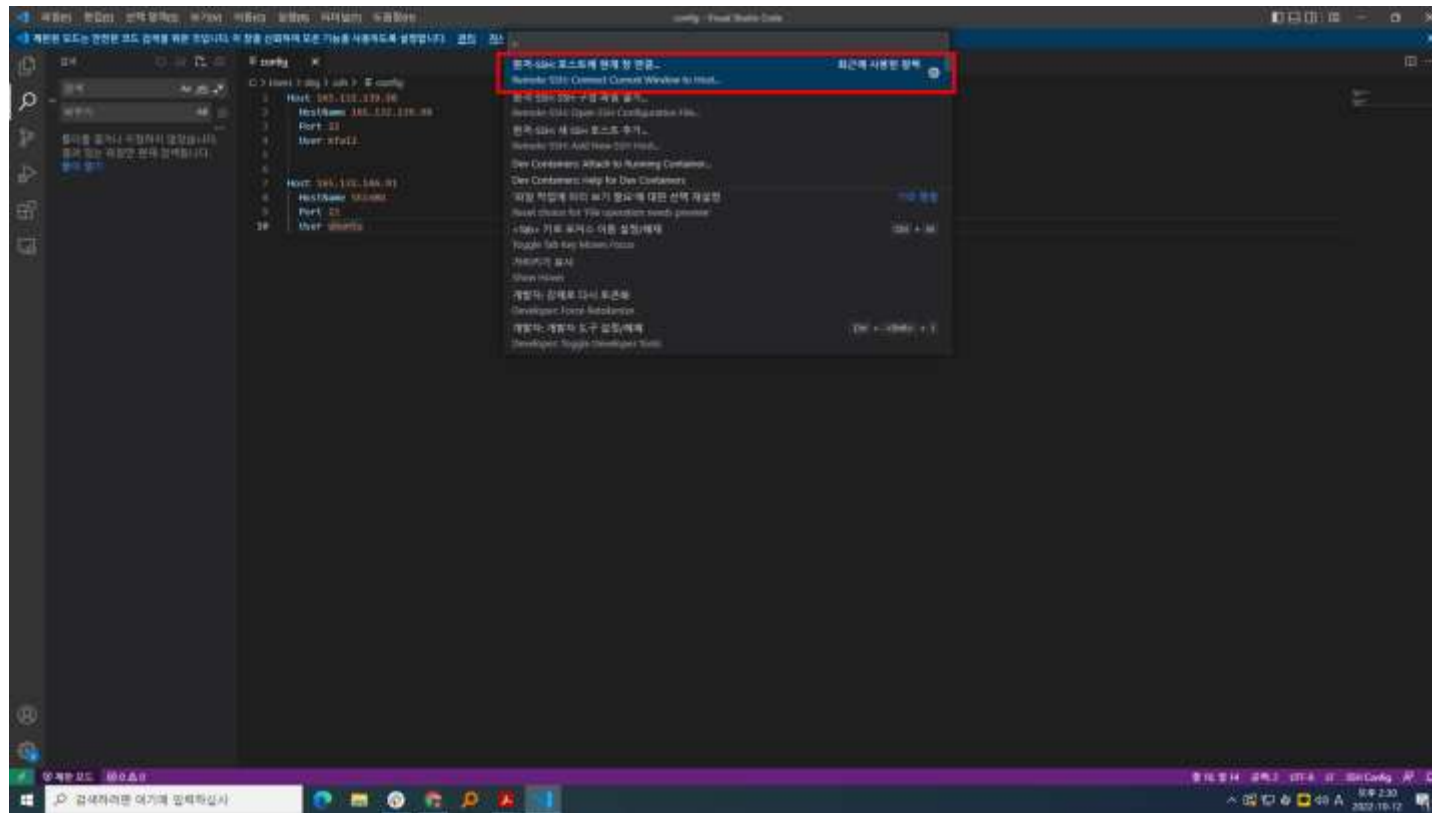
4. C:/Users/[로컬\_컴퓨터\_계정명]/.ssh/config 파일 클릭해서

```
C: > Users > username > .ssh > config
1 Host server
2   HostName 165.132.146.91
3   Port 22
4   User testuser
5
```

입력 후, 저장. Hostname은 고정되어 있으며, User는 추후 생성된 계정 이름으로 입력.  
Host는 임의로 지정 가능.

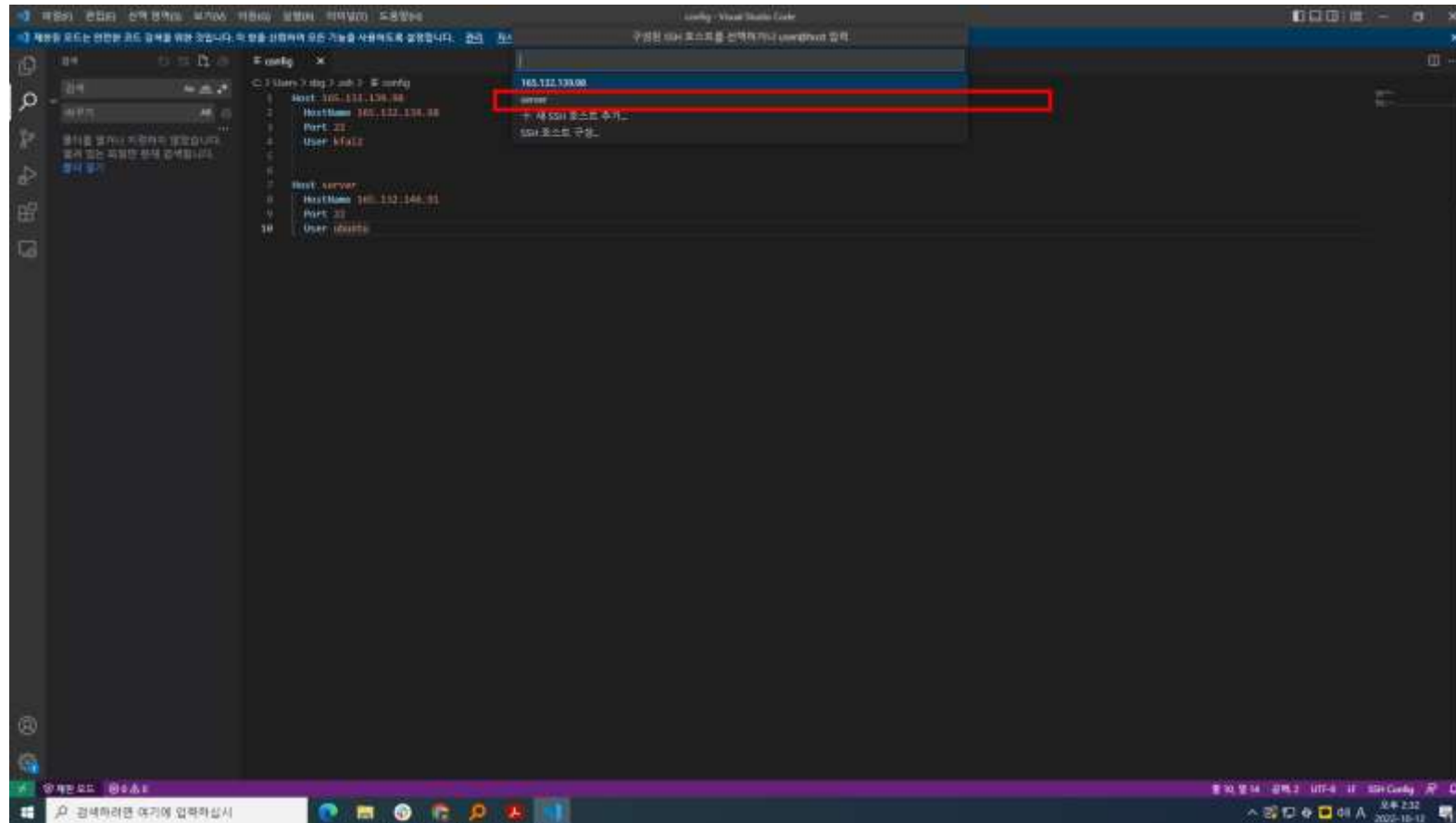
# SSH 연결

5. Ctrl + shift + p → remote-ssh 호스트에 현재 창 연결 클릭



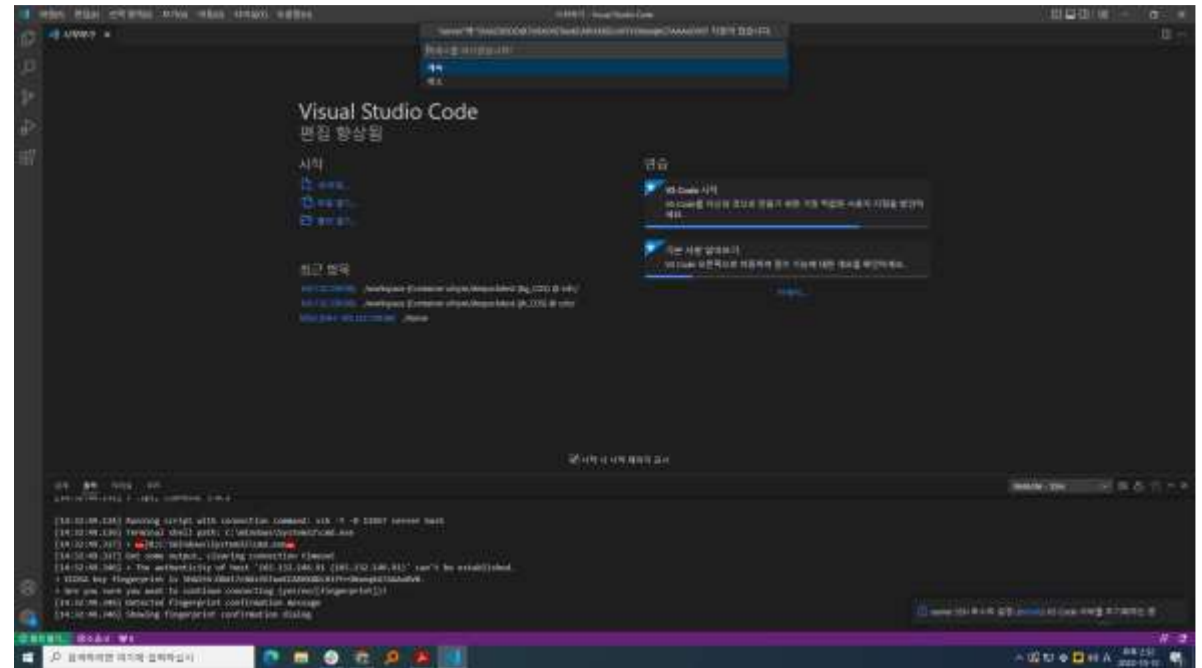
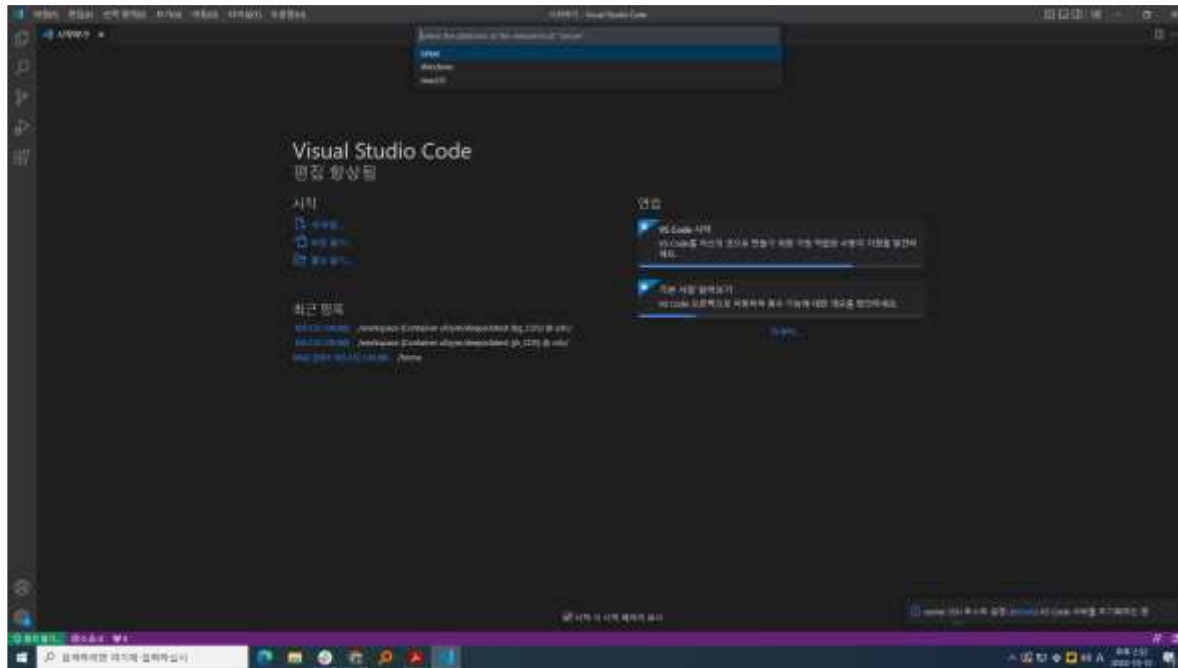
# SSH 연결

## 6. 목록에서 server 클릭



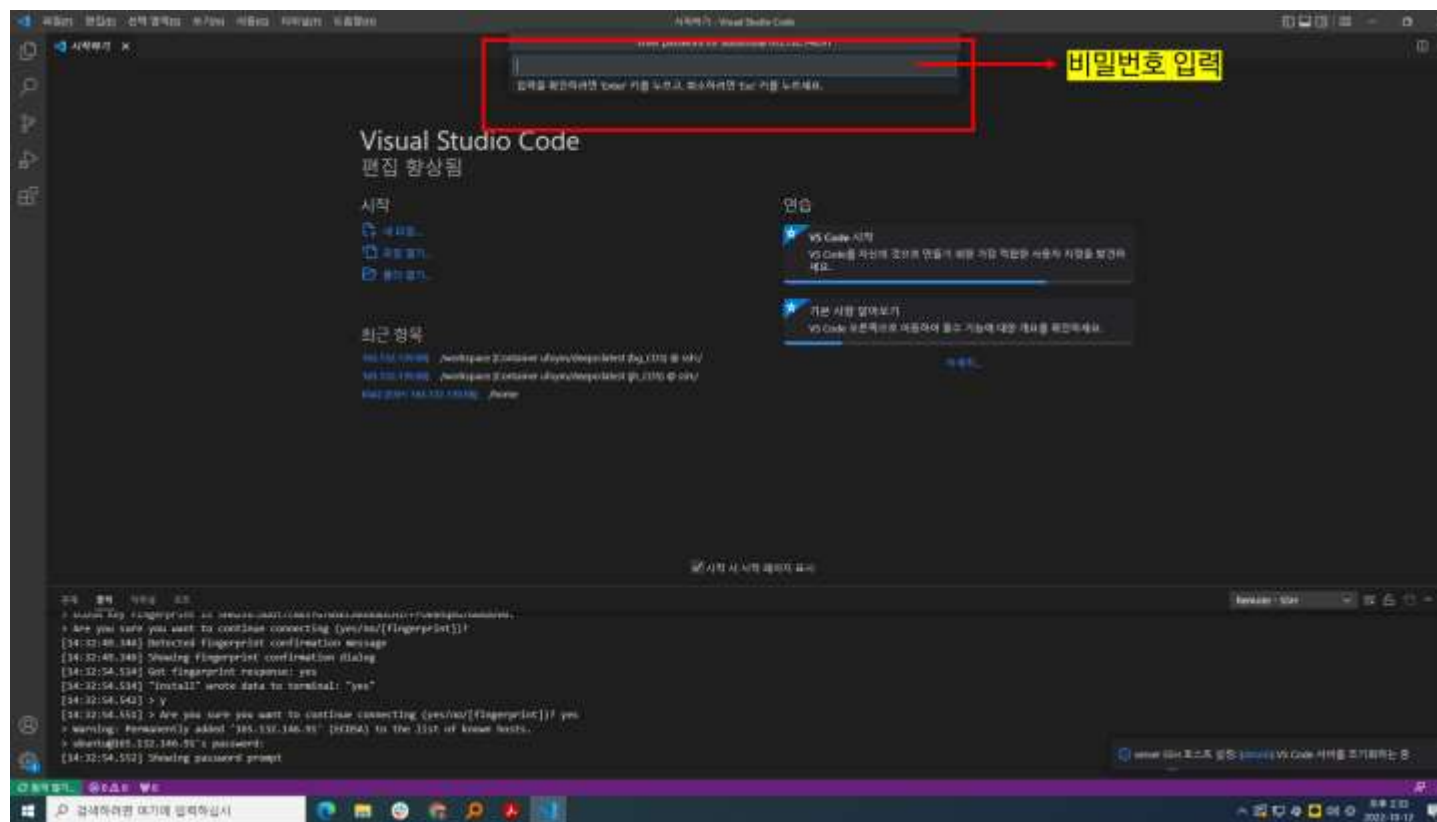
# SSH 연결

## 7. Linux → 계속



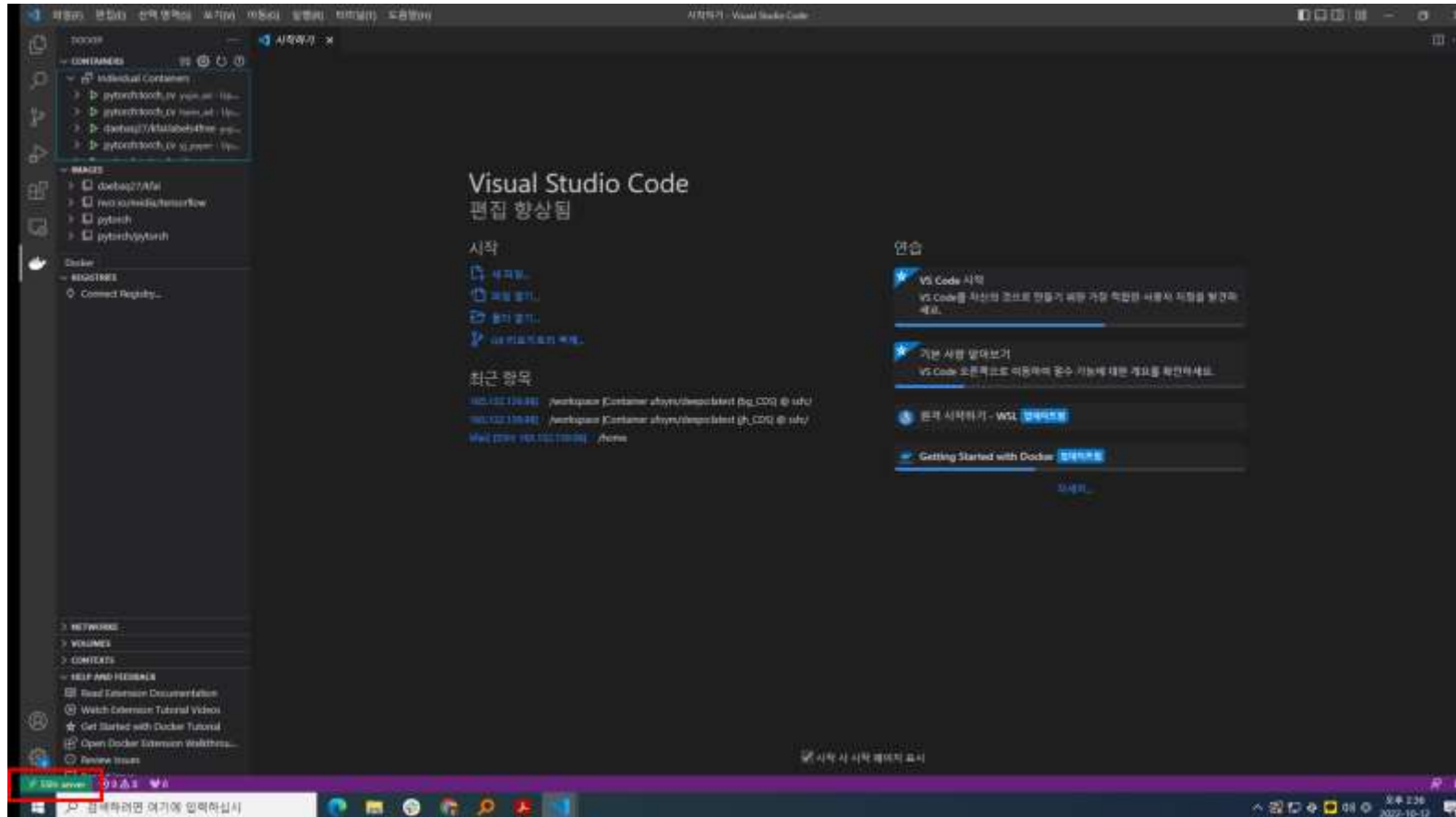
# SSH 연결

## 8. 비밀번호 입력 (비밀번호:testuser)



# SSH 연결

## 9. 접속 완료 (빨간 박스 친 부분에 SSH:server 라는 문구가 떠있는지 확인)



## 2. 도커 컨테이너 생성 (option)

# 도커 컨테이너 생성

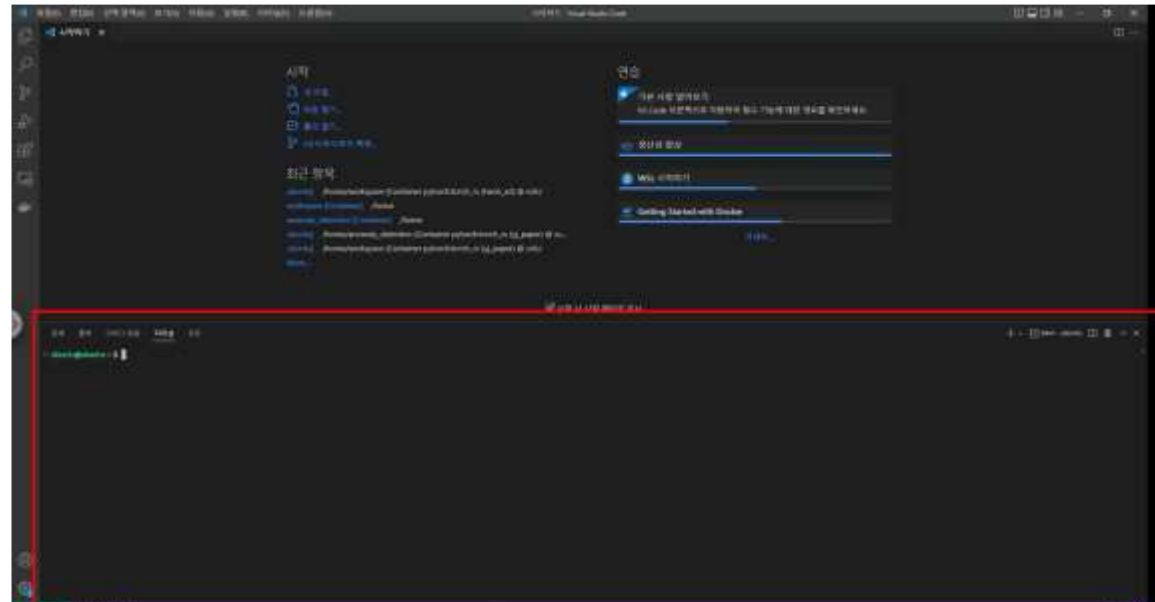
---

- 컨테이너 생성은 현재 조교가 진행하여 드리고 있으니 도커 컨테이너 생성 관련한 부분은 개별적으로 생성을 원하지 않는 한 참고하시지 않으셔도 좋습니다.



# 도커 컨테이너 생성

- 도커가 궁금하다면? <https://rfriend.tistory.com/744> 참고
  - 도커 컨테이너의 경우, 조교가 일괄적으로 생성해드릴 예정이나 원하는 도커 이미지가 따로 있을 경우 등을 고려하여 사용자가 직접 생성하여 사용도 가능합니다. 다만, 지정한 gpu device 이외에 타 device를 사용하시는 것이 적발될 경우 서버 이용에 불이익이 발생할 수 있습니다.
- 컨테이너 생성
  - Vscode에서 ctrl + ` 를 입력하여 터미널 열기 (빨간 박스 부분이 터미널)



# 도커 컨테이너 생성

- 터미널에 도커 컨테이너 생성 및 실행 명령어 입력하기

생성 명령어는 다음과 같습니다.

```
docker container run -it --gpus “device={지정된 device number}”  
--ipc=host --name [원하는 컨테이너 이름]  
--hostname [원하는 컨테이너 이름] -v ~/[작업하고자 하는 폴더]:[본인이 원하는 컨테이너 내 경로]  
-p 포트번호 ufoym/deepo:latest /bin/bash
```

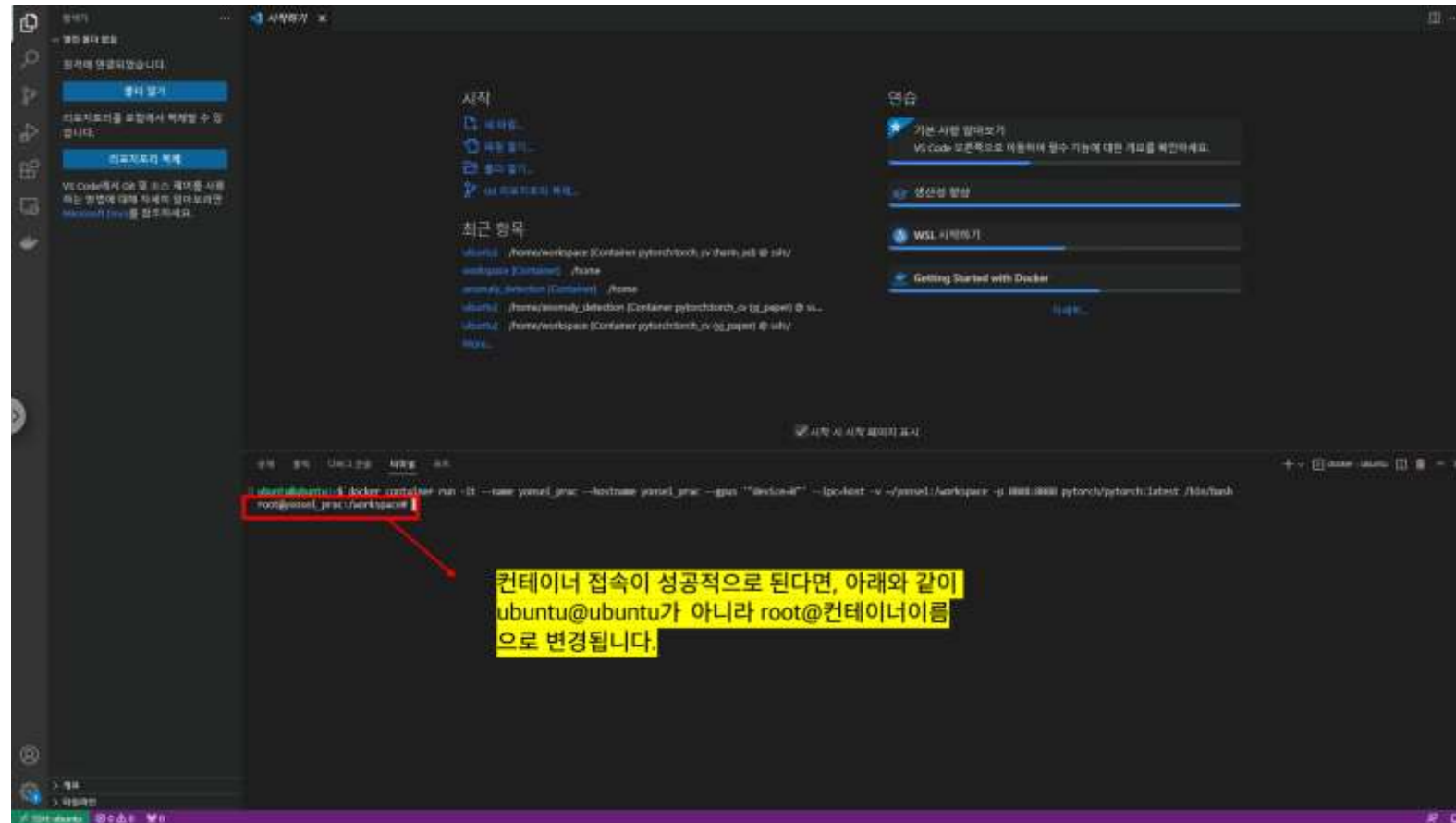
## 아래는 예시

```
docker container run -it --gpus '"device=0"'  
--ipc=host --name yonsei_prac  
--hostname yonsei_prac -v ~/testuser:/workspace  
-p 8080 ufoym/deepo:latest /bin/bash
```

→ 원하는 컨테이너 이름은 본인 이니셜 혹은 본인 영문 이름이 먼저 들어가도록.

# 도커 컨테이너 생성

- 생성 완료 시 화면



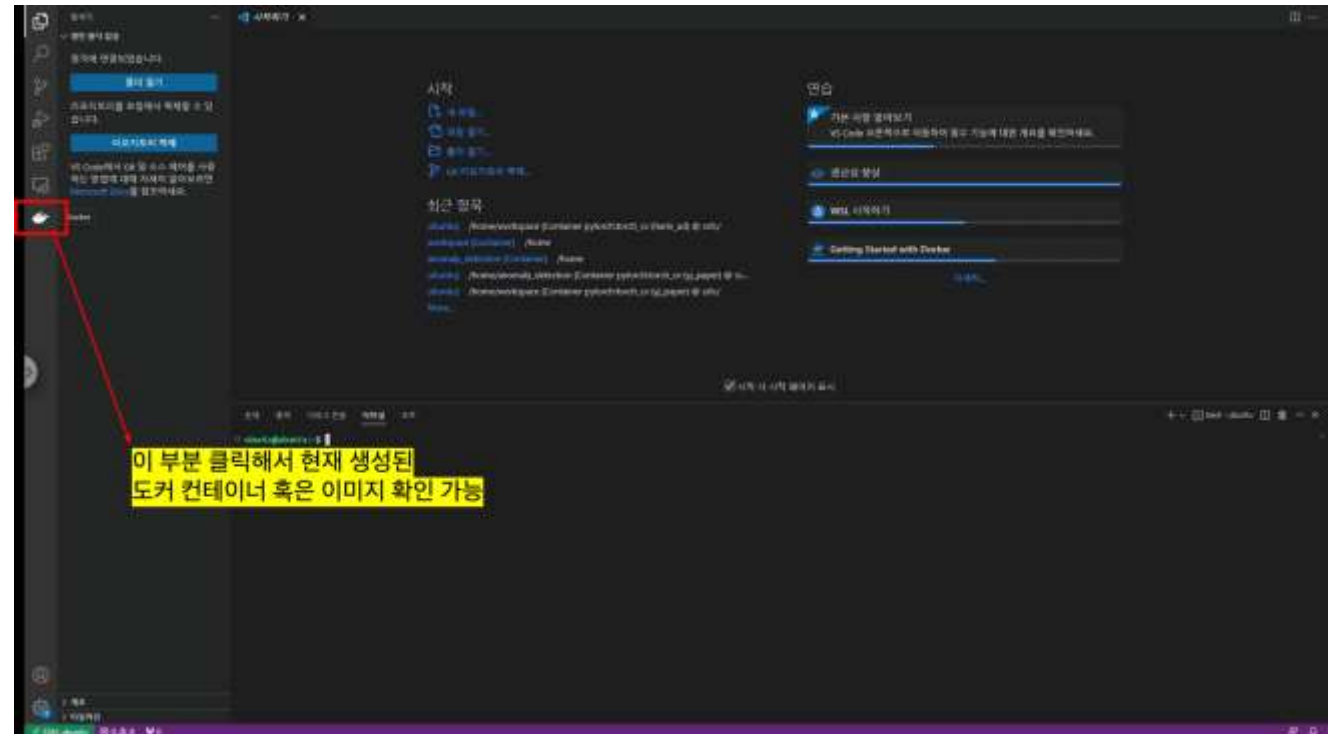
# 3. VSC에 컨테이너 연결

# 도커 컨테이너 연결

- 컨테이너 vscode에 연결하기

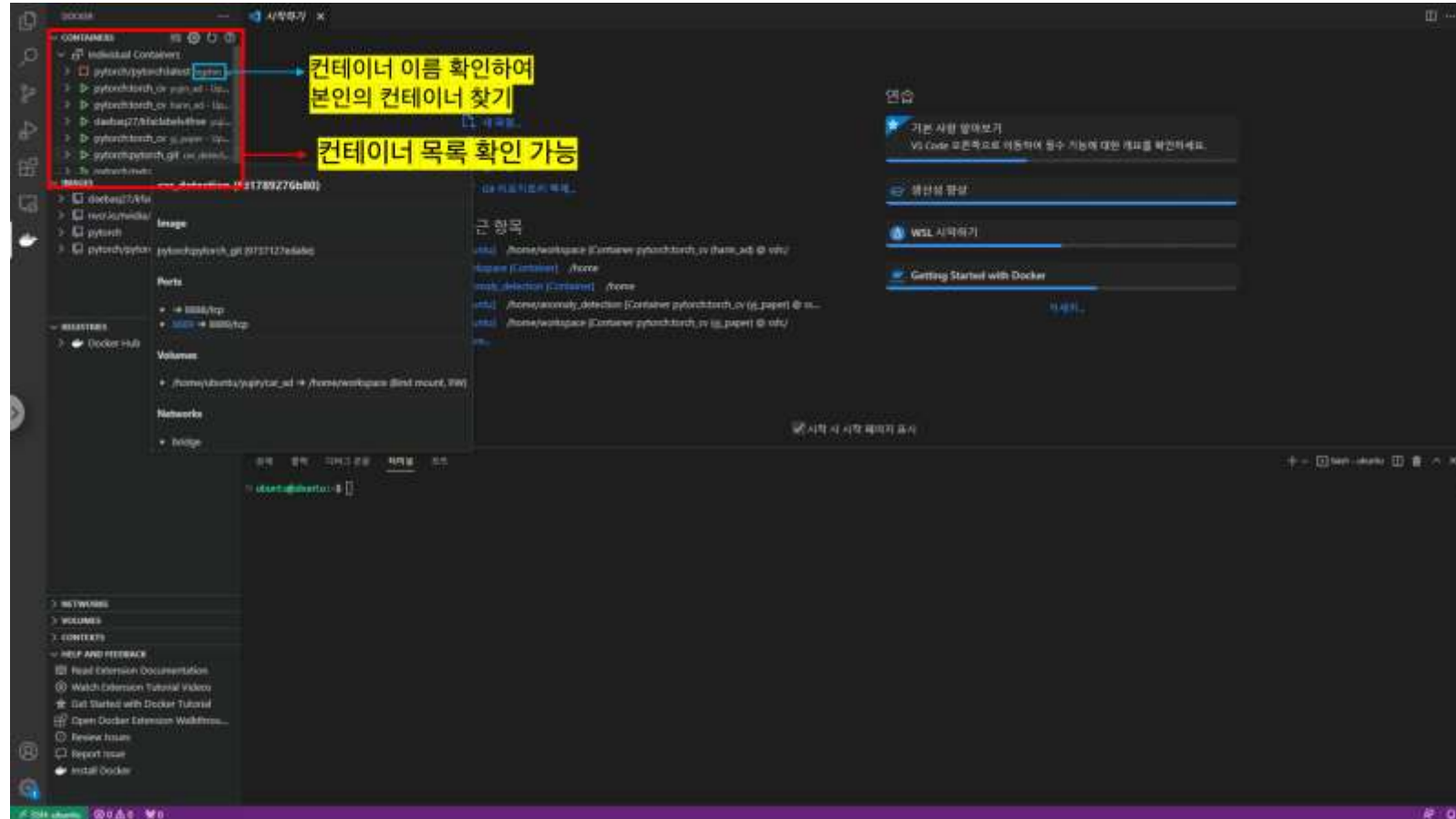
터미널보다는 vscode와 같은 IDE (통합개발환경) 에서 작업하는 게 더 수월합니다. 따라서, vscode 화면에서 컨테이너를 활용할 수 있도록 vscode에 컨테이너를 연결해주어야 합니다.

## 1. VSC 내 도커 관련 창으로 이동



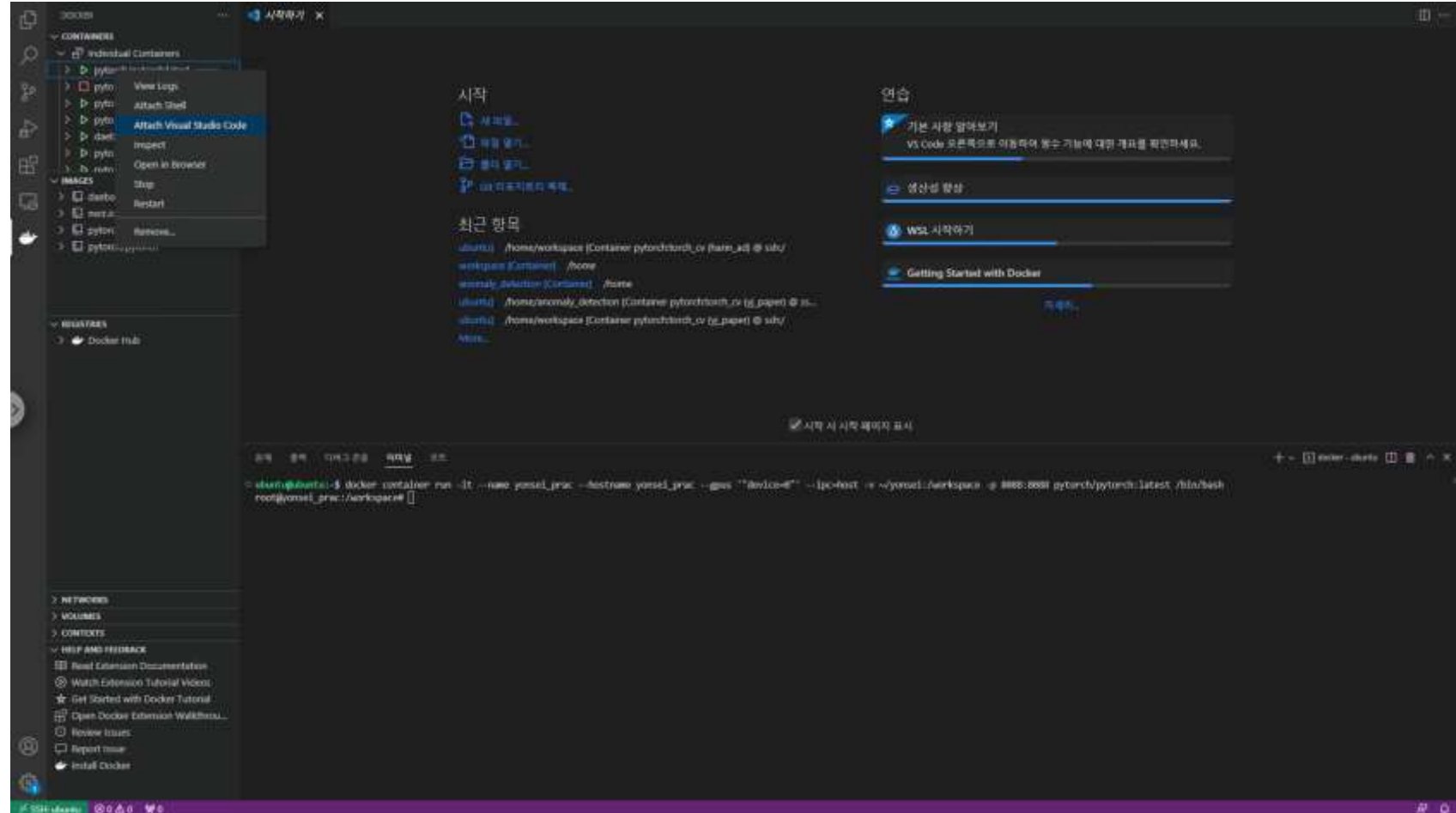
# 도커 컨테이너 연결

## 2. 본인의 도커 컨테이너 확인



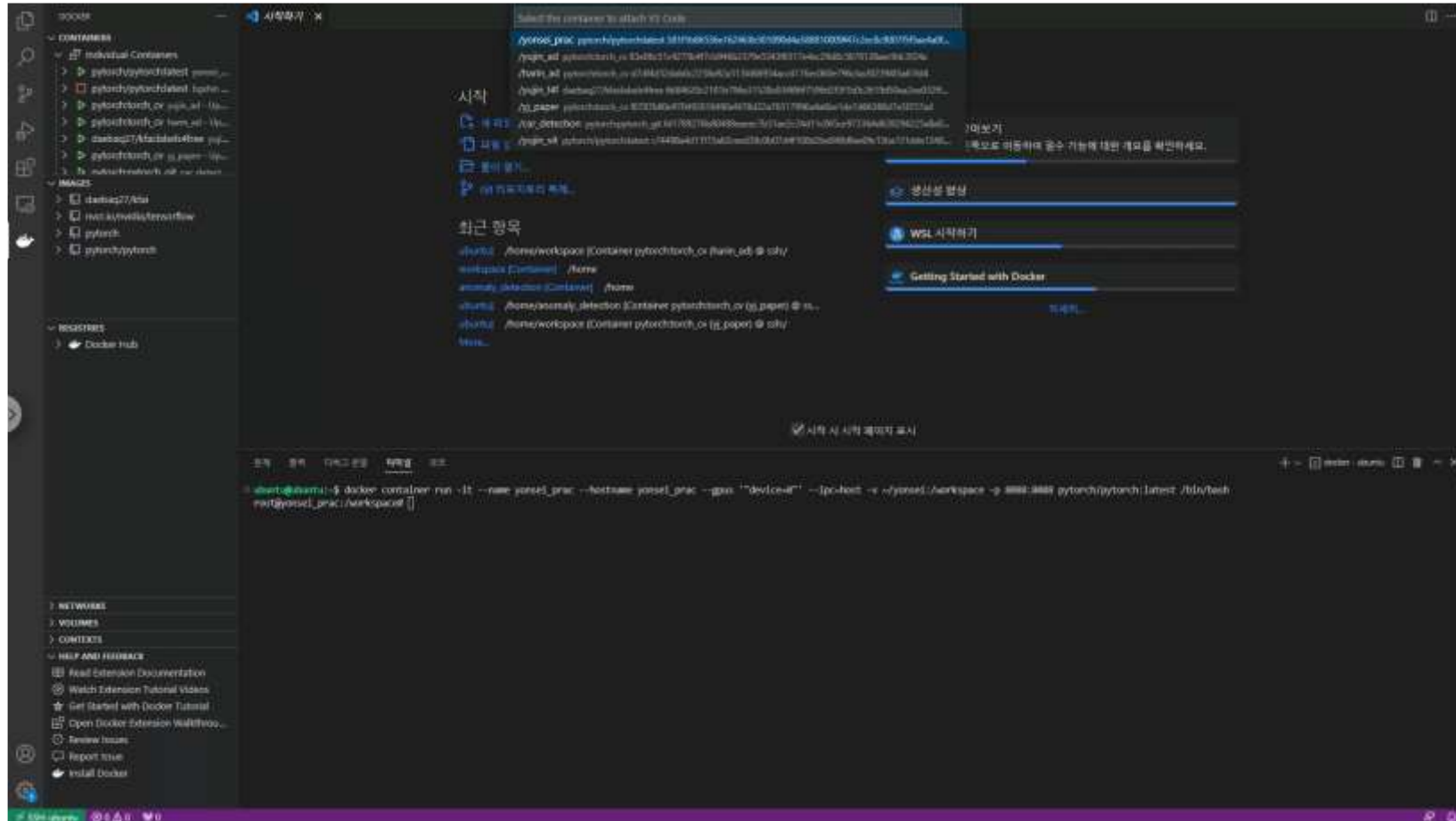
# 도커 컨테이너 연결

## 3. 본인 컨테이너 우클릭 후, Attach Visual Studio 클릭



# 도커 컨테이너 연결

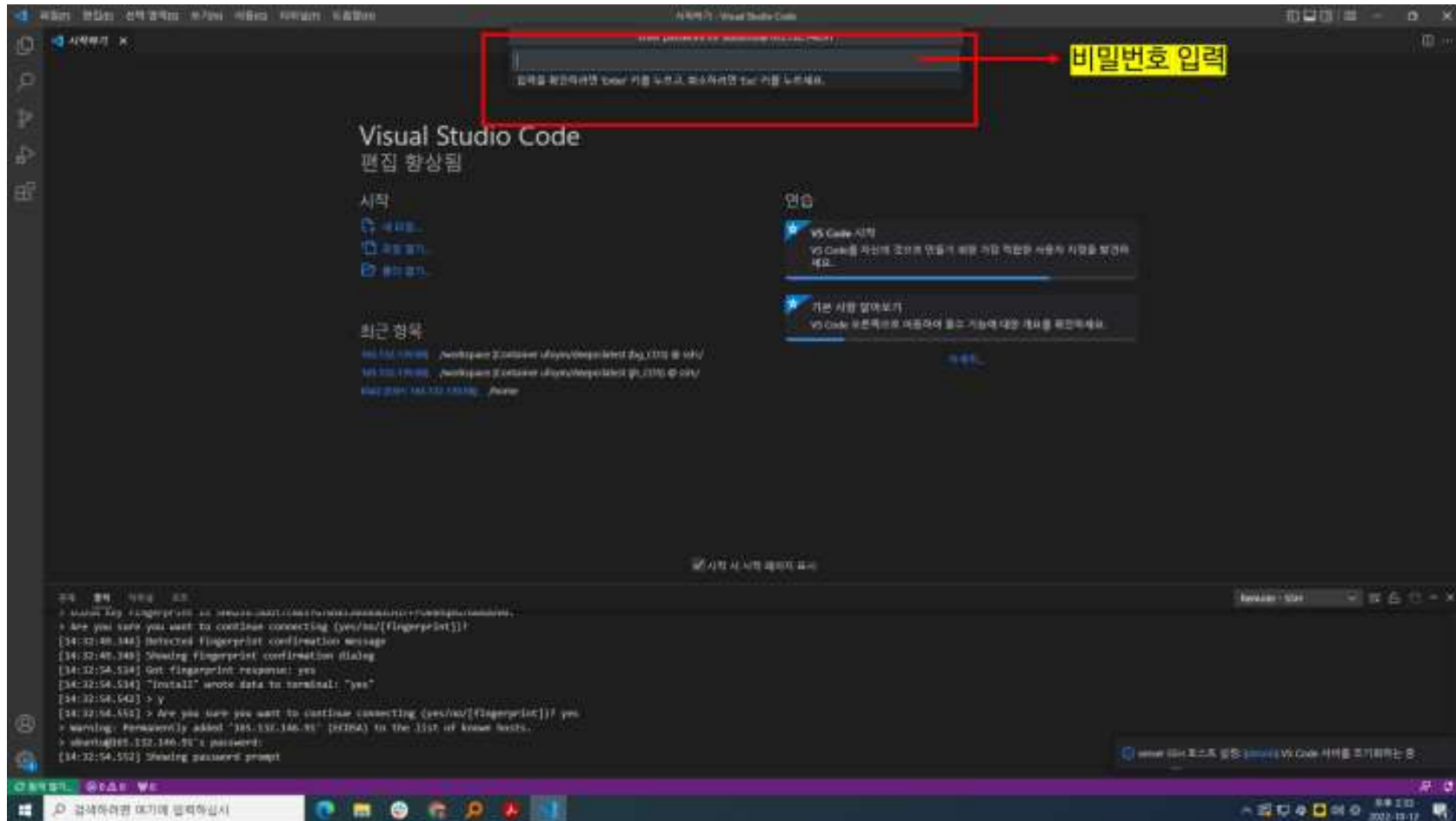
## 4. 창이 뜨면 본인 컨테이너 클릭





## 도커 컨테이너 연결

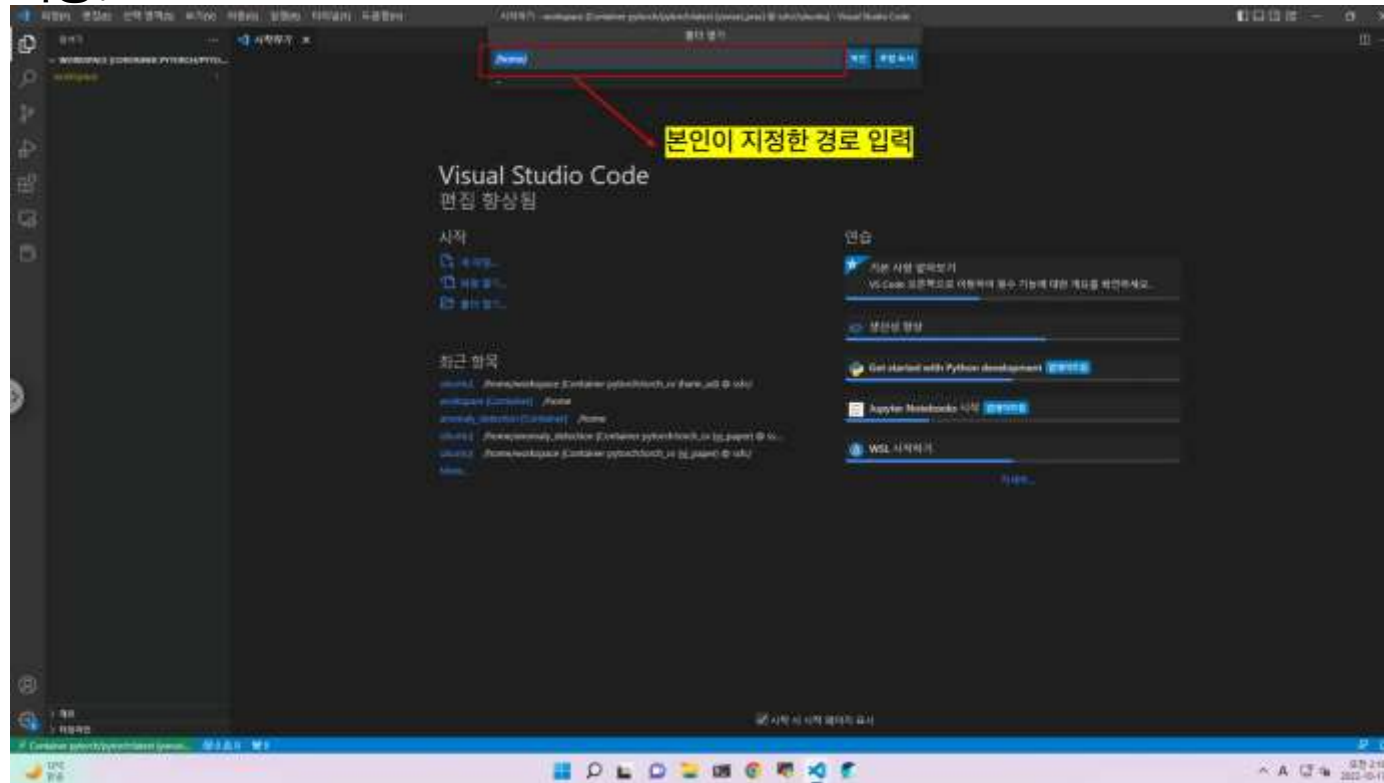
5. 비밀번호 입력 (서버 비밀번호와 동일) 후, 접속



# 도커 컨테이너 연결

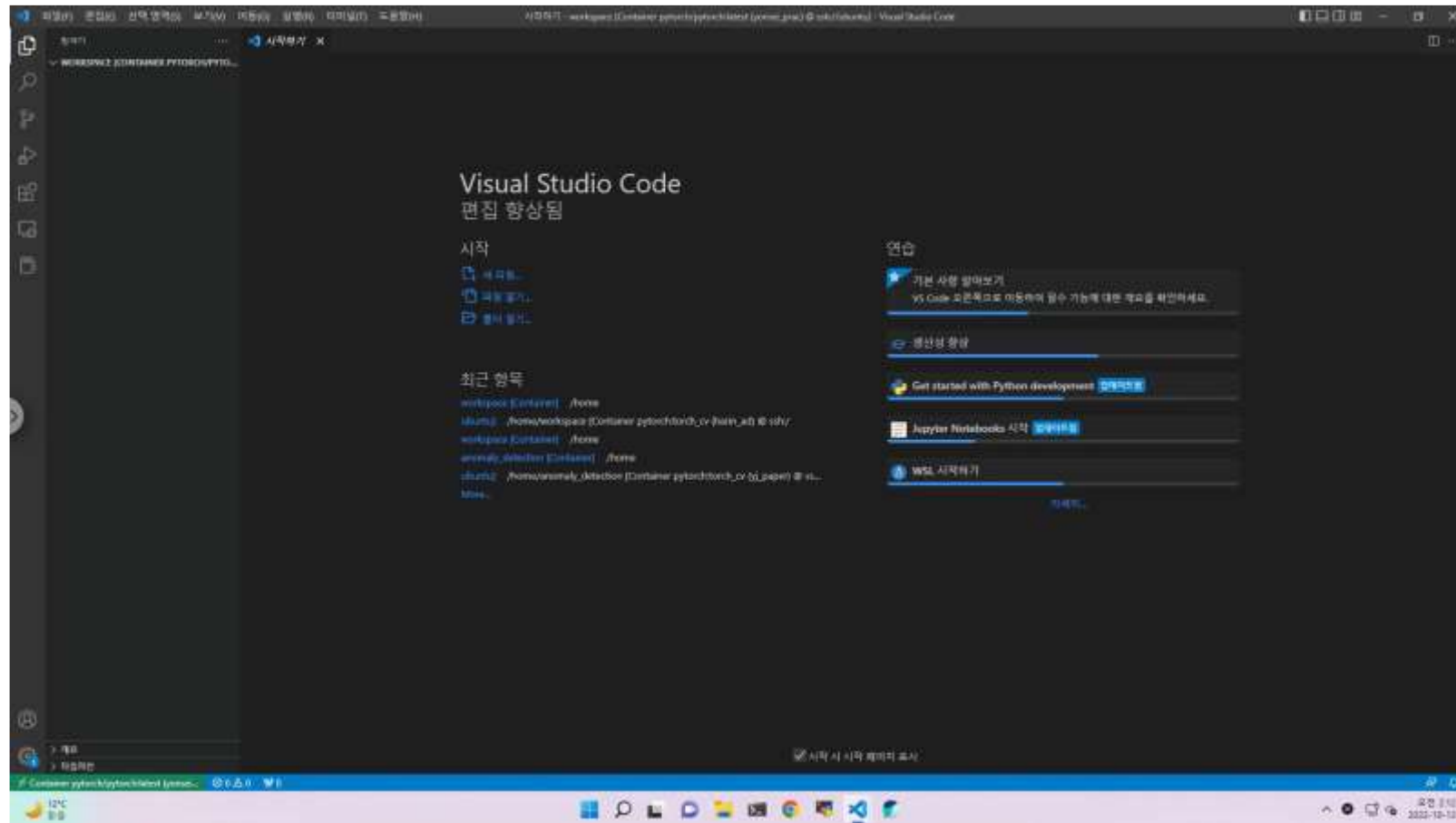
## 5. 비밀번호 입력 후, 접속 (ctd.)

만약, 접속 시 작업 영역이 존재하지 않는다고 하면 작업 영역 열기를 클릭하여 본인이 지정한 컨테이너 내 작업 공간 경로로 이동.



# 도커 컨테이너 연결

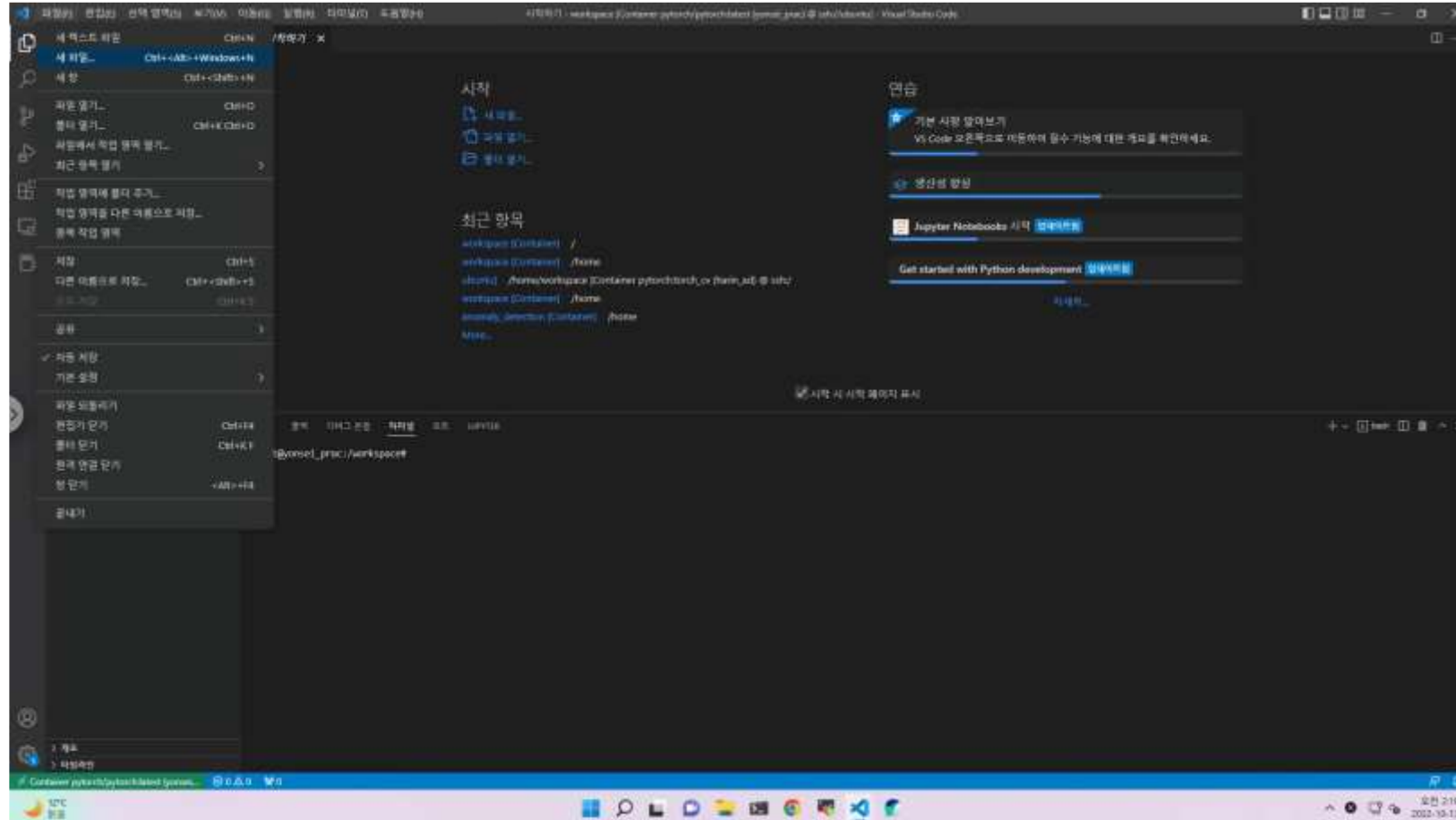
## 6. 접속 완료



# 4. 컨테이너에서 코드 작업

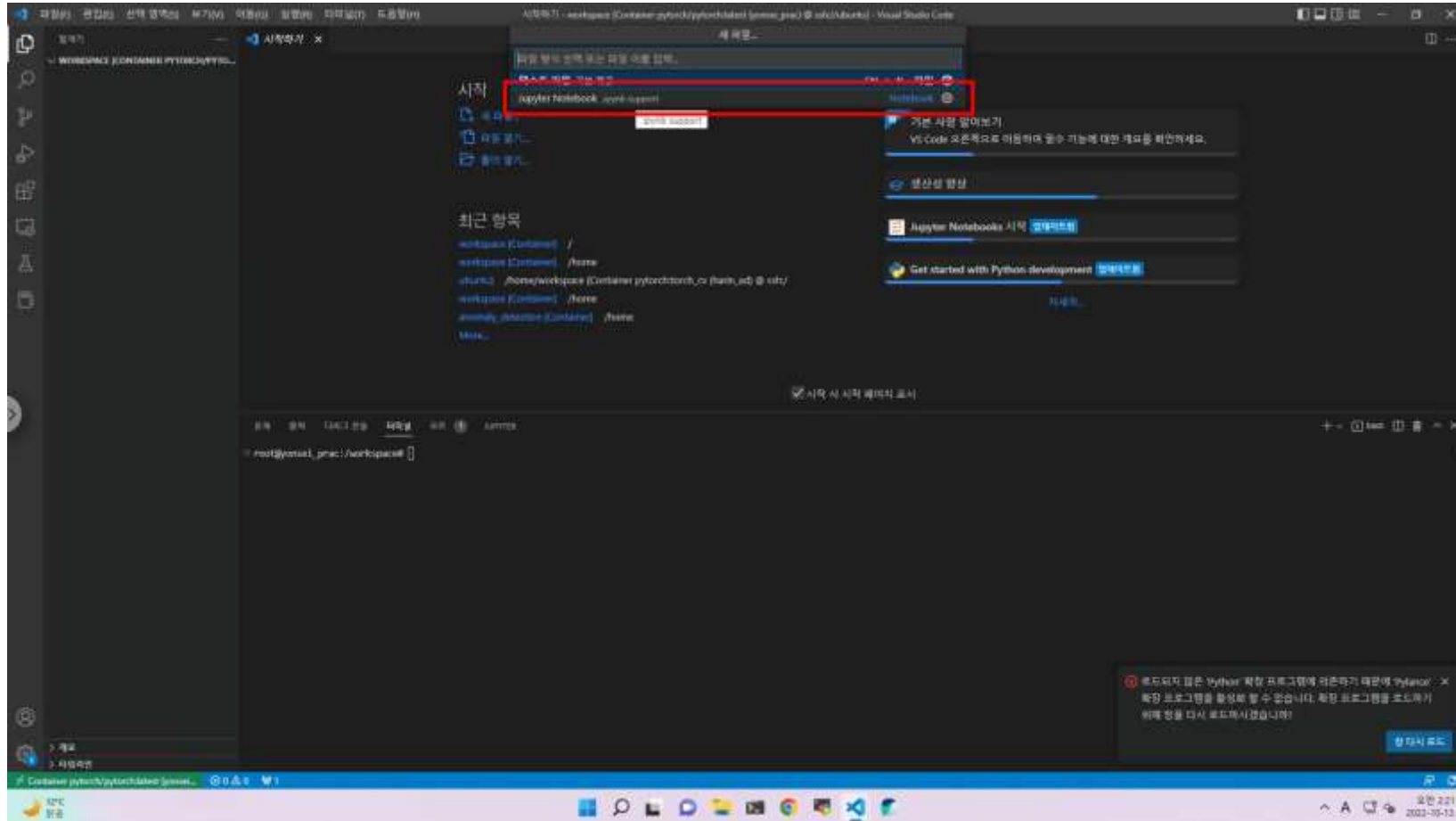
# 컨테이너에서 주피터노트북 파일 생성 후 작업

1. 좌상단의 ‘파일’ 클릭 후, ‘새 파일’ 클릭. 혹은 Ctrl+Alt+Windows+N 단축키 입력



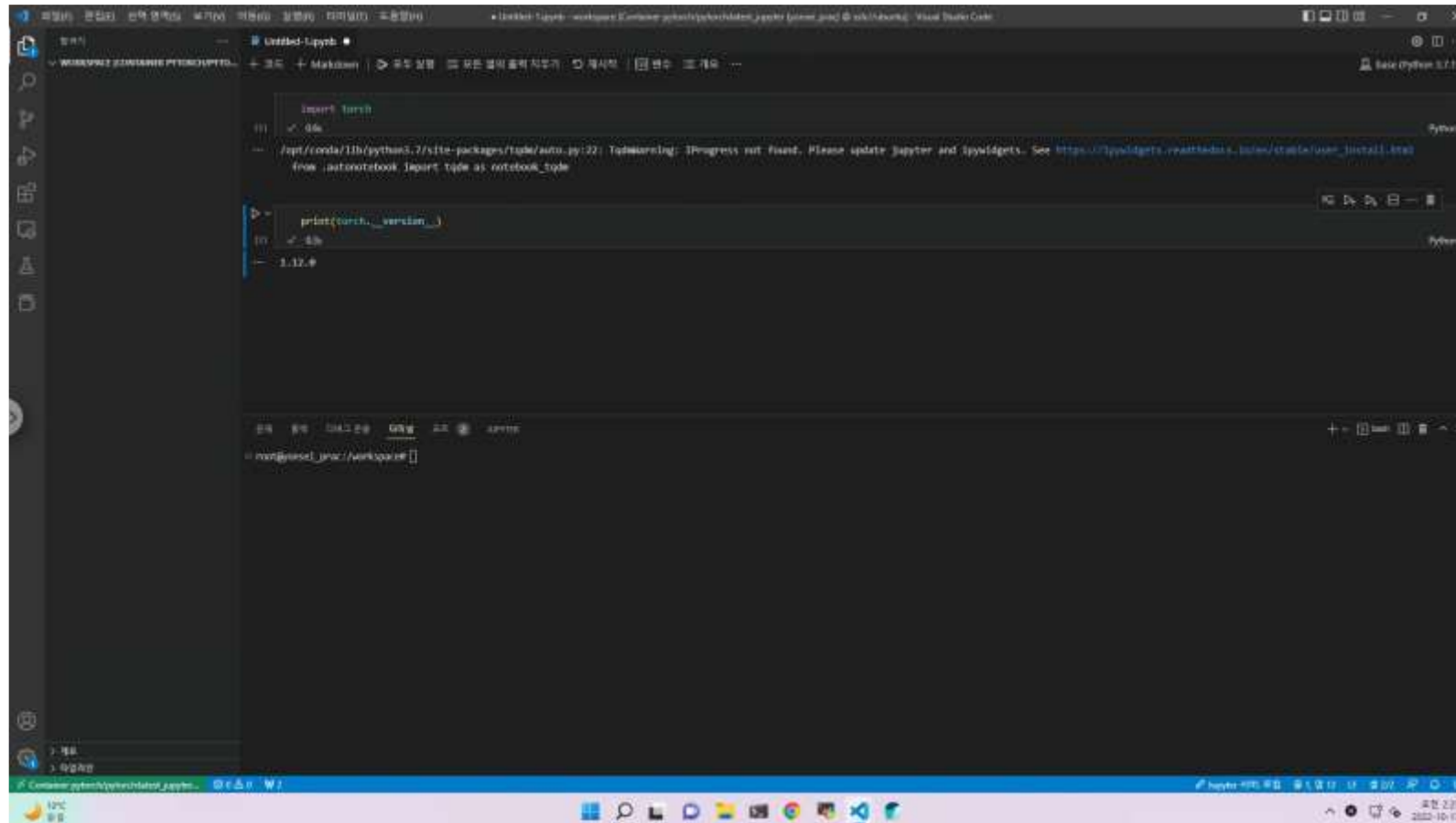
# 컨테이너에서 주피터노트북 파일 생성 후 작업

## 2. 주피터 노트북 클릭



# 컨테이너에서 주피터노트북 파일 생성 후 작업

## 3. 주피터 노트북 파일 생성 후 결과 (import torch and print version)



The screenshot displays a Jupyter Notebook environment within a container. The notebook has two cells. The first cell contains the code `import torch` and has executed successfully, returning `<module 'torch' from '/opt/conda/lib/python3.7/site-packages/torch/__init__.py'>`. The second cell contains the code `print(torch.__version__)` and has also executed successfully, returning the string `1.12.0`. The interface includes a file explorer on the left, a command prompt at the bottom, and a Windows taskbar at the very bottom.

```
import torch
<module 'torch' from '/opt/conda/lib/python3.7/site-packages/torch/__init__.py'>

print(torch.__version__)
1.12.0
```

*Knowledge Fusion and Artificial Intelligence Lab*

지식융합 인공지능 연구실

@ Yonsei University

**감사합니다.**