



# Urban Bigdata and AI Institute of University of Seoul

---

슈퍼컴퓨터 사용법 교육

# 01 교육프로그램 및 UBAI 소개

## 교육 자료 다운로드

- [https://github.com/uos-ubai/transit\\_seoul](https://github.com/uos-ubai/transit_seoul)
- 해커톤\_슈퍼컴퓨터 사용법 교육.pdf와 edu.txt 그리고 requirements.txt를 다운 받아주세요.

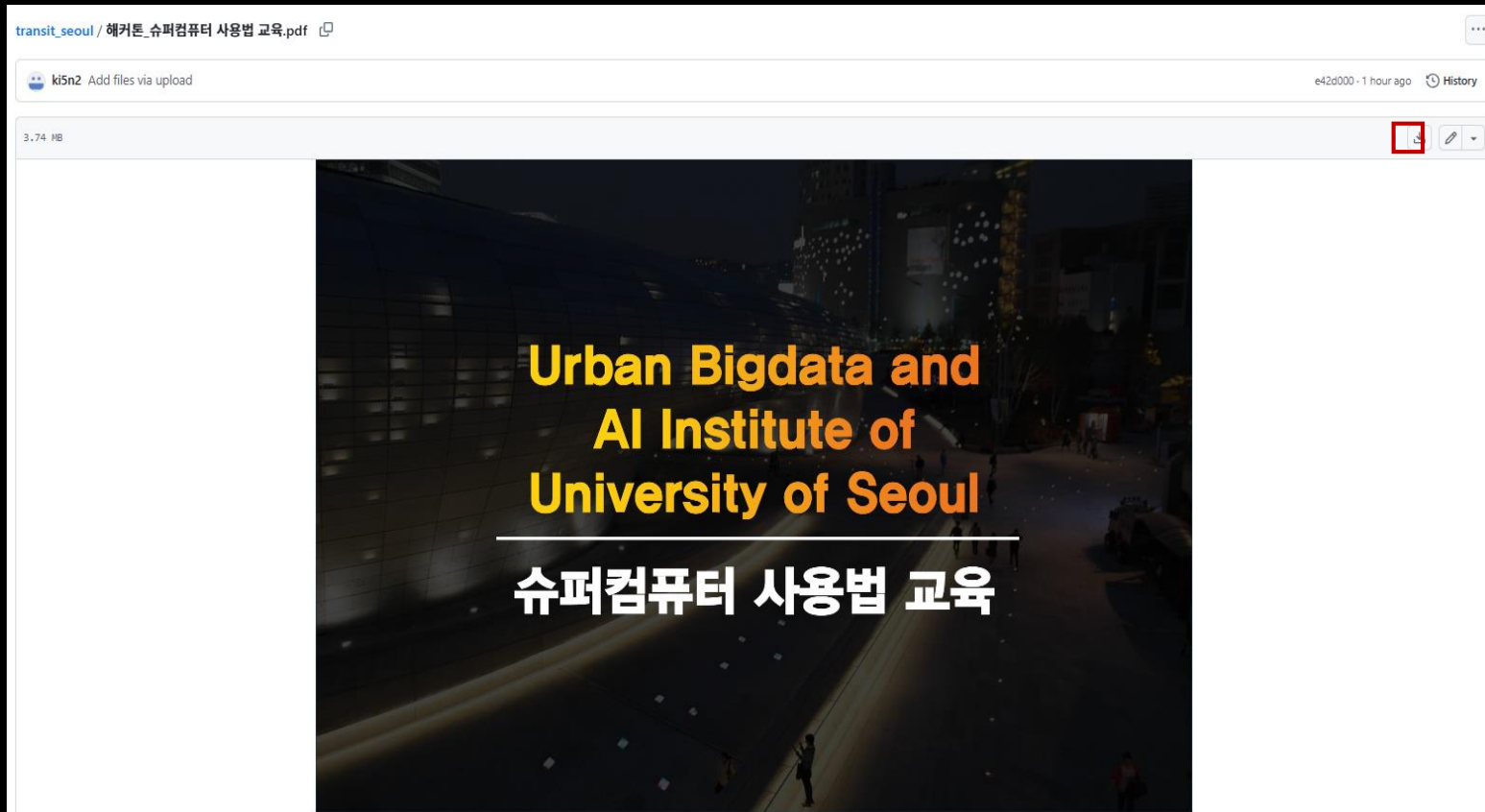
The screenshot shows the GitHub interface for the repository 'transit\_seoul' by user 'ki5n2'. The repository is public and has 1 branch and 0 tags. The file list includes:

File Name	Commit Message	Time Ago
README.md	Update README.md	5 hours ago
edu.txt	Add files via upload	8 minutes ago
requirements.txt	Add files via upload	13 minutes ago
run.sh	Create run.sh	5 hours ago
test.py	Add files via upload	7 minutes ago
transit_seoul_sample_code_2.py	Add files via upload	3 minutes ago
transit_seoul_sample_code_3.py	Add files via upload	3 minutes ago
해커톤_슈퍼컴퓨터 사용법 교육.pdf	Add files via upload	now

# 01 교육프로그램 및 UBAI 소개

## 교육 자료 다운로드

- [https://github.com/uos-ubai/transit\\_seoul](https://github.com/uos-ubai/transit_seoul)
- 해커톤\_슈퍼컴퓨터 사용법 교육.pdf와 edu.txt 그리고 requirements.txt를 다운 받아주세요.



# 01 교육프로그램 및 UBAI 소개

## 교육 목차

시작 시간	종료 시간	소요 시간	교육 내용
10:00	10:10	10min	교육 프로그램 및 UBAI 소개
10:10	10:40	30min	UBAI 서버 접속
10:40	11:00	20min	Python 환경 구축 (Miniconda)
11:00	11:10	10min	쉬는 시간
11:10	12:00	50min	슈퍼컴퓨팅 환경 이해 및 Job 제출

- WIFI: UBAI\_CO\_SPACE\_01\_5G/6G 또는 UBAI\_CO\_SPACE\_02\_5G/6G 로 접속
  - PW: *bigdata2025*

# 01 교육프로그램 및 UBAI 소개

## 슈퍼컴퓨터란 무엇인가?

- 슈퍼컴퓨터(Supercomputer)는 일반 컴퓨터보다 훨씬 빠른 속도로 방대한 계산을 수행할 수 있는 고성능 컴퓨터(HPC, High Performance Computing)입니다.
- 일반 PC가 1개의 CPU로 일처리를 한다면, 슈퍼컴퓨터는 수천~수십만 개의 CPU, 또는 GPU를 동시에 사용하여 병렬로 작업을 처리합니다.
- 주로 기후 및 분자 시뮬레이션, 유전체 분석, 인공지능 모델 학습 등과 같은 대규모 과학 및 공학 문제 해결에 사용됩니다.

## 도시과학빅데이터 · AI연구원(UBAI) 운영방향

### 01 도시과학 연구

---

도시 경쟁력을 갖춘 선도도시의 기반이 되는  
도시과학 연구

### 02 인재양성

---

빅데이터와 인공지능 분야의  
역량을 갖춘 인재 양성

### 03 시스템 구축

---

창업 및 신규 첨단산업 생태계 구축  
최첨단 빅데이터 · 인공지능 기술 개발 및 인프라 제공

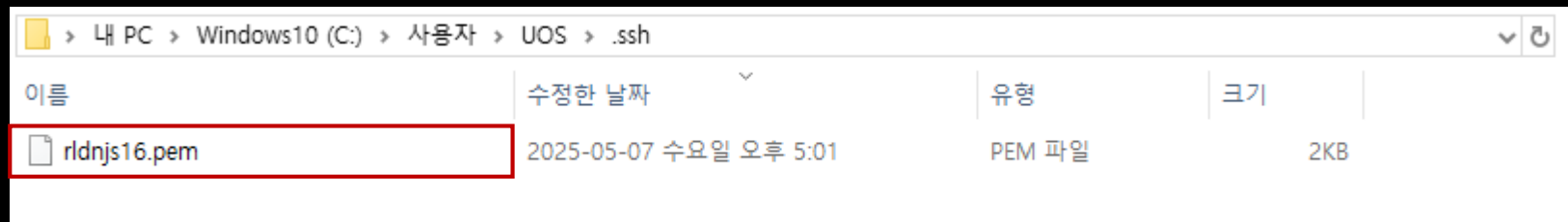
### 04 대 · 내외 협력

---

서울 및 국내외 기관과의 대내외 협력

### 계정 생성

- UBAI Cluster 사용을 위해서는 팀별 계정을 발급받아야 합니다.
- 키 파일을 받은 상태에서 실습에 참가해야 합니다.
  - 동일한 키 파일이라도, 여러 대의 노트북에서 접속 가능
  - 즉, 팀장이 받은 키 파일을 통해 팀원 노트북에서 접속 가능

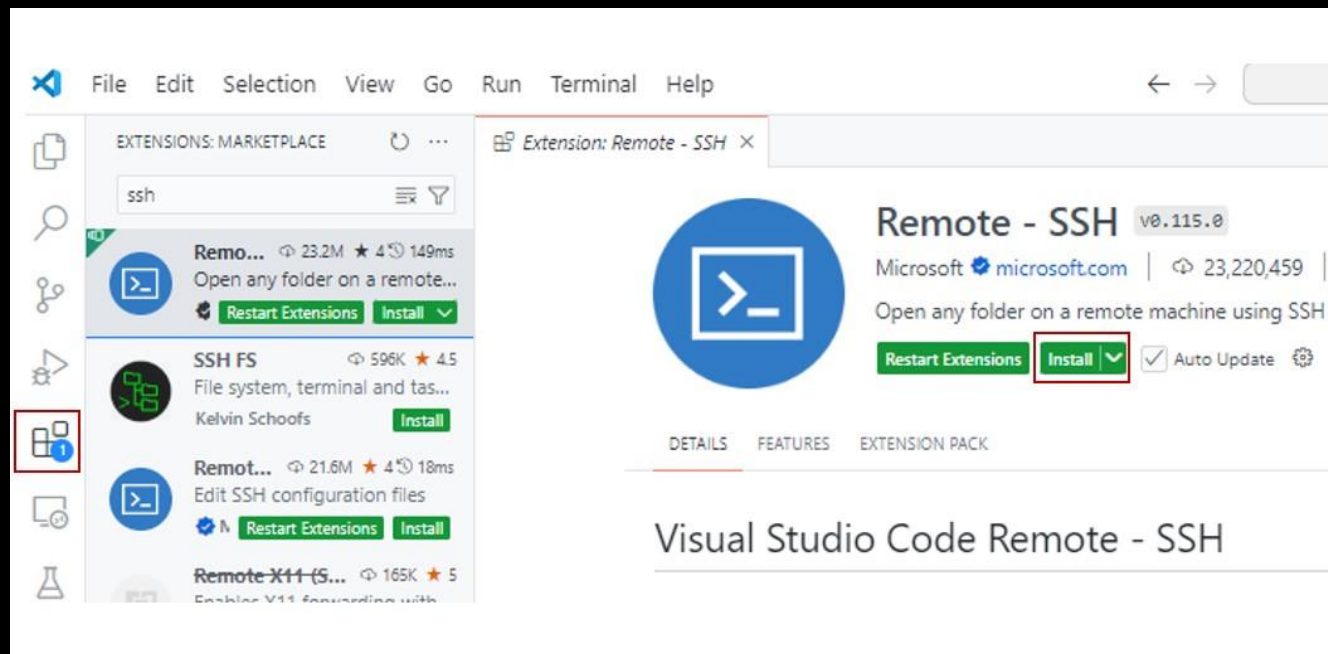


내 PC > Windows10 (C:) > 사용자 > UOS > .ssh				
이름	수정한 날짜	유형	크기	
rldnjs16.pem	2025-05-07 수요일 오후 5:01	PEM 파일	2KB	

*C:\User\{사용자이름폴더}\.ssh\*

### 접속 방법

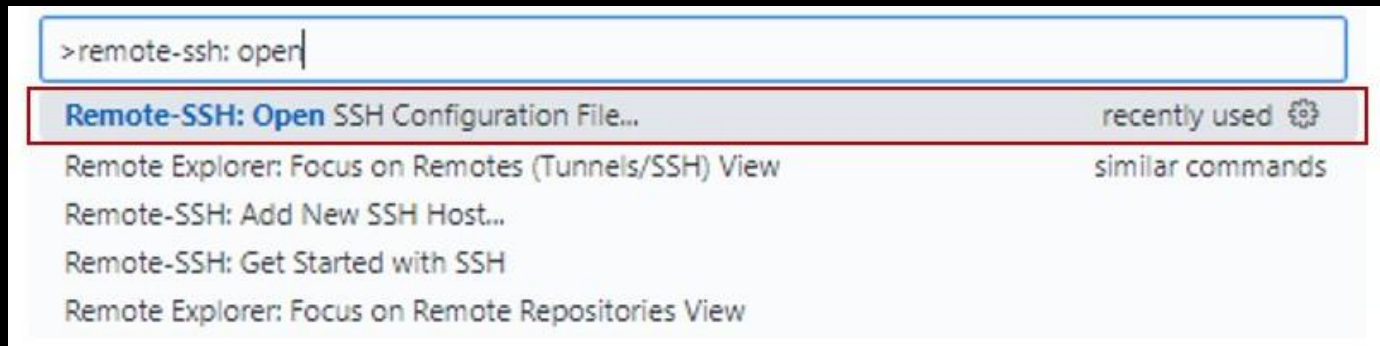
- Visual Studio Code 다운로드 및 접속
  - <https://code.visualstudio.com/download>
- Remote-SSH Extension 설치





### 접속 방법

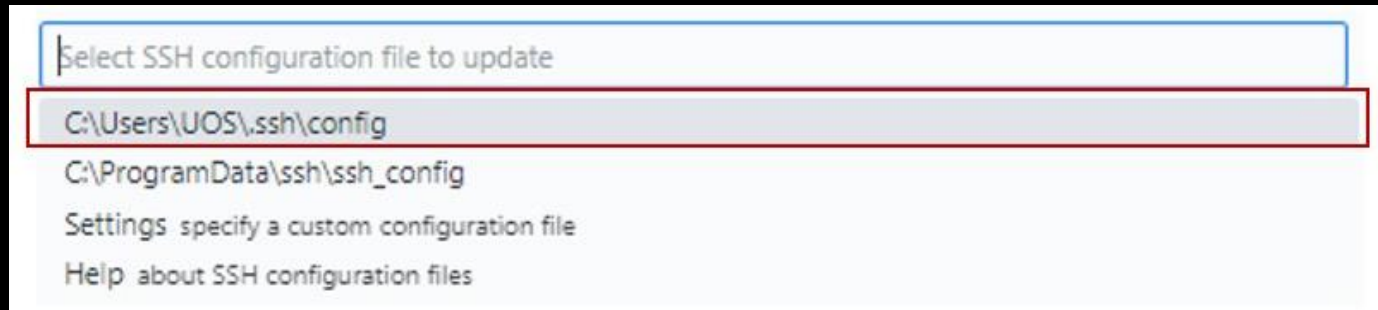
- Config 파일 수정
  - Ctrl + Shift + P
  - Remote-SSH: Open SSH Configuration File...



- .ssh폴더가 없는 경우에는 .ssh폴더와 config파일을 같이 생성
- .ssh폴더가 있는 경우에는 config파일만 생성

### 접속 방법

- Config 파일 수정
  - 이후, 입력창에 C:\Users\사용자\.ssh\config 선택



### 접속 방법

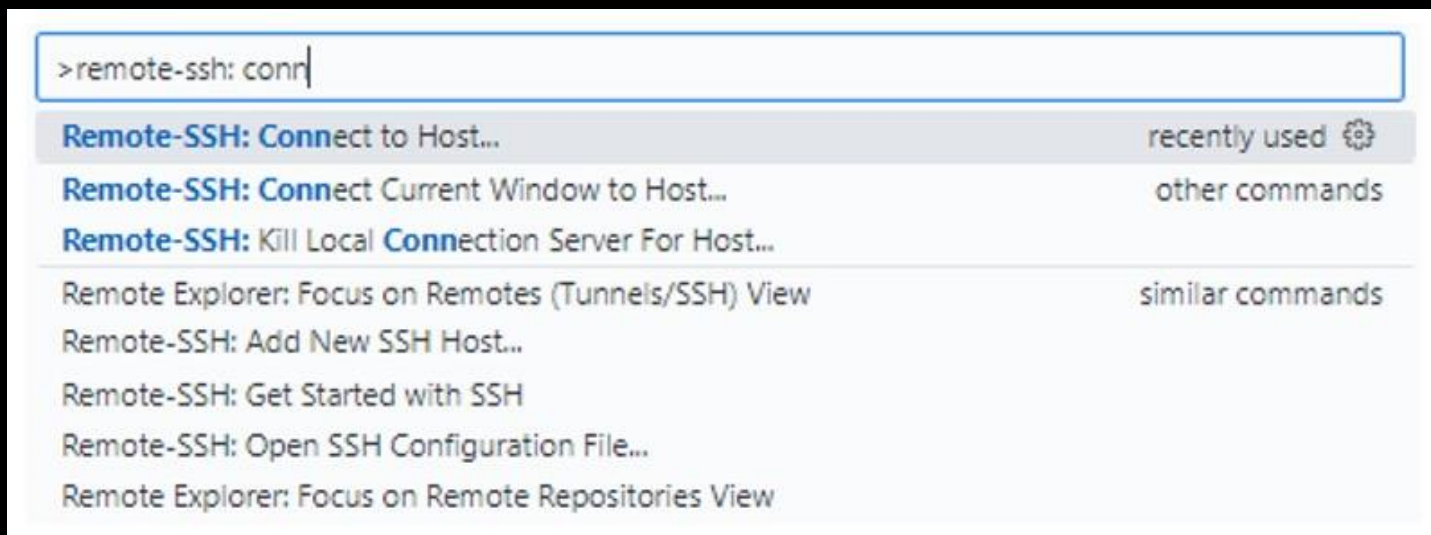
- Config 파일 수정

```
7 Host gate2
8   |   HostName 172.16.10.37
9   |   Port 22
10  |   User rldnjs16
11  |   IdentityFile C:\Users\UOS\.ssh\rldnjs16.pem
```

- Host: gate2
- HostName:
  - 172.16.10.37
  - 반드시 해당 경로로 접속
- Port: 22
- User: 사용자ID
- IdentityFile: 키파일 경로
  - Windows : C:\Users\W[사용자 이름]\.ssh\key.pem
  - MacOS : /Users/[사용자 이름]/.ssh/key.pem

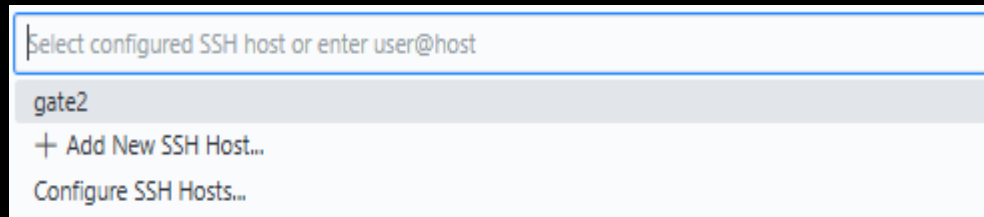
### 접속 방법

- SSH 접속
  - Ctrl + Shift + P
  - > Remote-SSH: Connect to Host...



### 접속 방법

- SSH 접속
  - Gate2로 접속

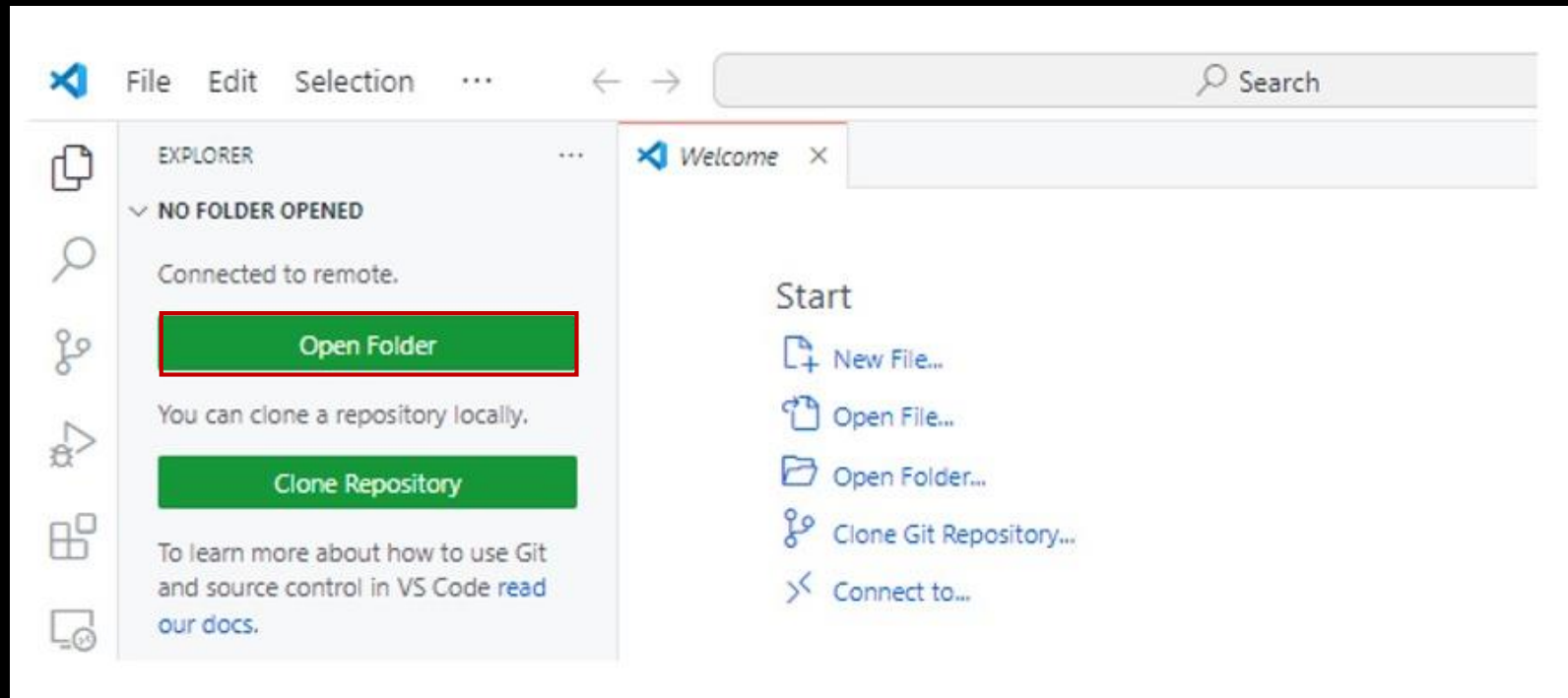


- 운영체제 **Linux** 선택(★ ★ ★)
- **Continue** 선택
- Gate 접속 확인 (왼쪽 하단)



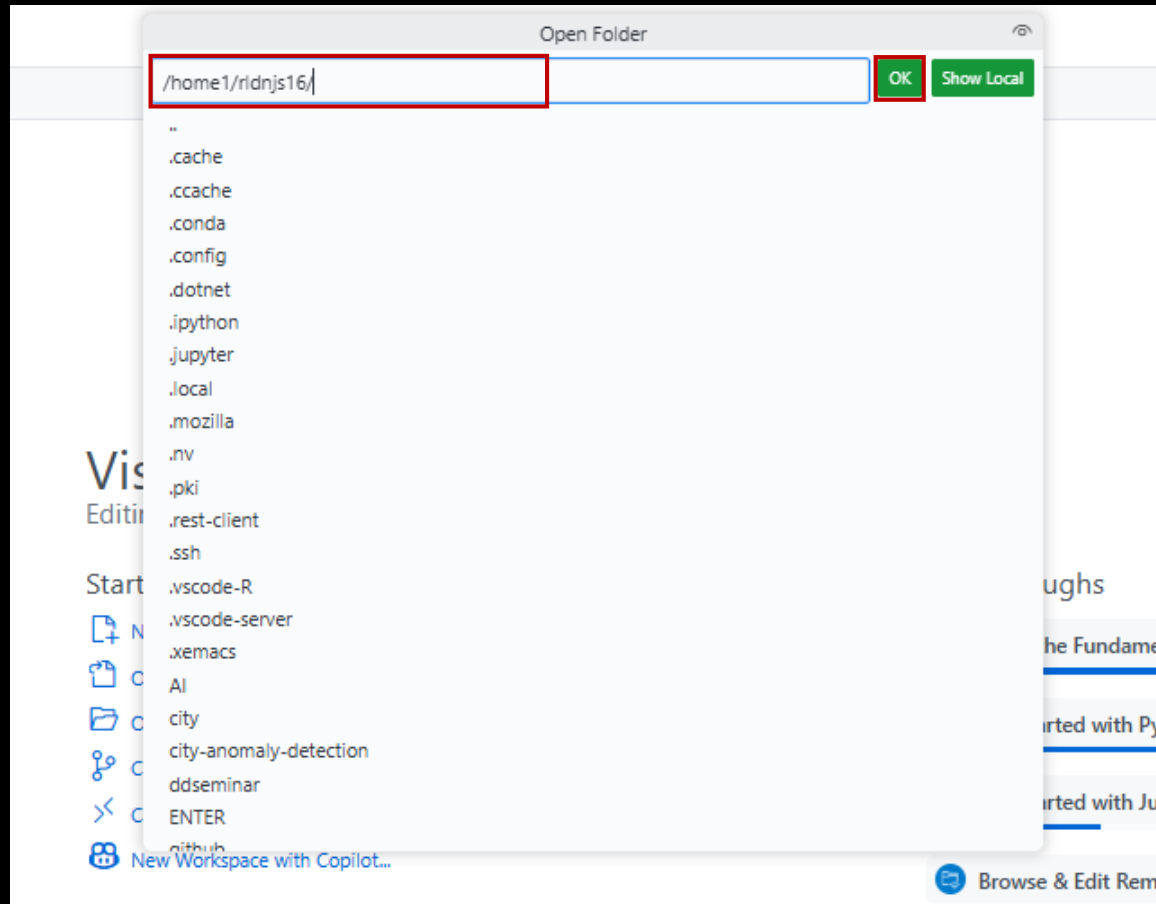
### 접속 방법

- 디렉토리 접속



### 접속 방법

- 디렉토리 접속
  - 팀 계정 폴더로 접속합니다.



### Python

- Python은 웹 어플리케이션, 소프트웨어 개발, 데이터 사이언스, 머신러닝, 딥러닝에 널리 사용되는 프로그래밍 언어입니다.
- 오픈소스 환경을 가지고 있으며, 많은 사람들이 이용하는 언어입니다.
- Python을 활용하기 위해서는 보통 Anaconda를 활용하지만 리눅스 컴퓨팅 환경에서는 Miniconda를 활용합니다.
- UBAI 슈퍼컴퓨터에서도 Python을 이용하기 위해서는 Miniconda를 사용해야 합니다.





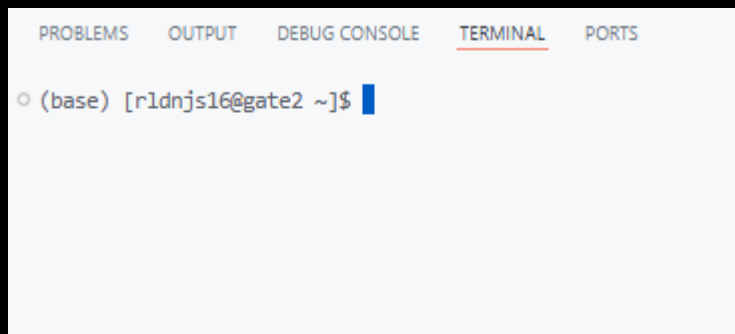
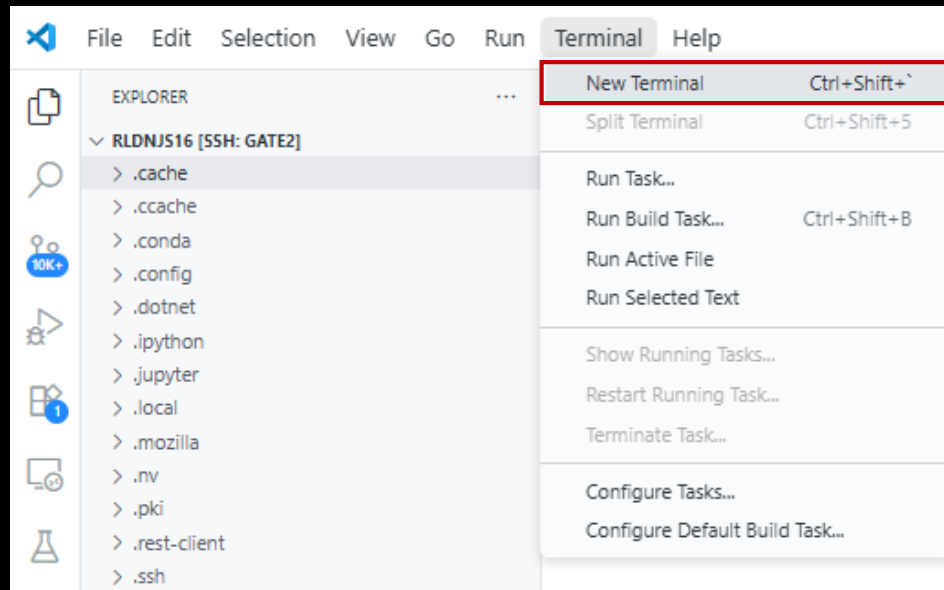
### Miniconda

- Anaconda는 머신러닝이나 데이터 분석 등에 사용하는 여러가지 패키지가 기본적으로 포함되어있는 파이썬 배포판입니다.
- Python의 가상환경을 구축하는데 매우 유용하게 사용됩니다.
- Miniconda는 Anaconda의 경량 버전이라고 볼 수 있습니다.



## Miniconda 설치

- Terminal 접속



### Miniconda 설치

- Miniconda 설치 파일 다운로드
  - 앞서 github에서 다운로드한 edu.txt에 저장된 명령어를 복사 후 붙여 넣어줍니다.
  - `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ (base) [rldnjs16@gate2 ~]$ https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

- Bash 명령어를 통해 Miniconda 설치
  - `bash Miniconda3-latest-Linux-x86_64.sh`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ (base) [rldnjs16@gate2 ~]$ bash Miniconda3-latest-Linux-x86_64.sh
```

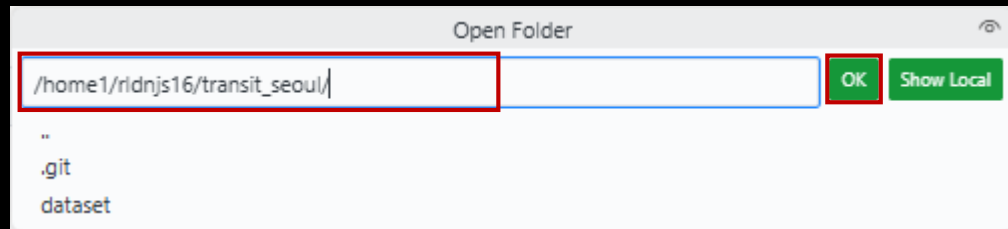
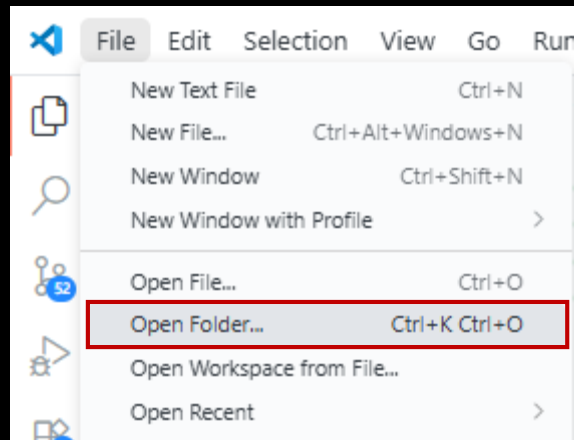
- 라이선스 동의에 대한 내용이 모두 나타날 때까지 **Enter**를 꼭 눌러줍니다.
- 그 후 라이선스 동의 확인에 대한 질문에 **yes**라고 입력해줍니다.
- Enter를 너무 오래 눌러 동의 **화면이 바로 넘어가지 않게** 주의해주세요.

### Miniconda 설치

- Bash 명령어를 통해 Miniconda 설치
  - 화면에서 해당 경로가 자신의 서버에 있는 경로와 맞는지 확인 후 Enter를 눌러줍니다.
  - 처음 접속 시, conda init 진행 선택에 대한 질문이 나타납니다.
  - Yes를 입력하신 후 enter를 눌러 주시면 됩니다.
  - 변경사항 반영을 위해 지금 작업한 창을 닫고 새로 접속해주세요.
  - 재접속 시, termina에 (base)[사용자ID@사용자\_gate2]가 보인다면 성공적으로 설치가 완료된 것입니다.
  - 설치 후 왼쪽의 탐색기(Explorer) 목록에 miniconda 폴더가 있는지 꼭 확인해주세요.

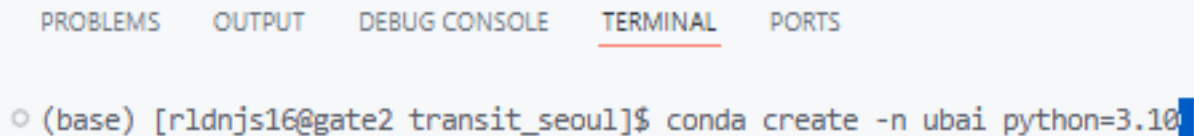
### Miniconda 실행

- 팀 계정 폴더 내 transit\_seoul 폴더로 이동합니다.



## Miniconda 실행

- 가상 환경 생성
  - `conda create -n {가상환경_이름} python={설치할_python_version}`

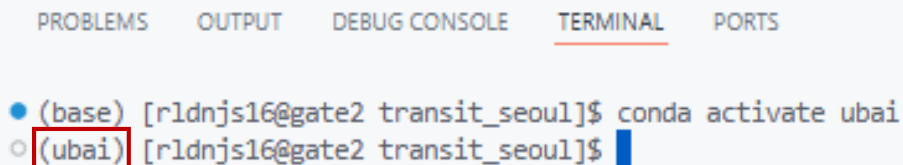


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
○ (base) [rldnjs16@gate2 transit_seoul]$ conda create -n ubai python=3.10
```

- Proceed ([y]/n)? y

- 가상 환경 실행
  - `conda activate {가상환경 이름}`

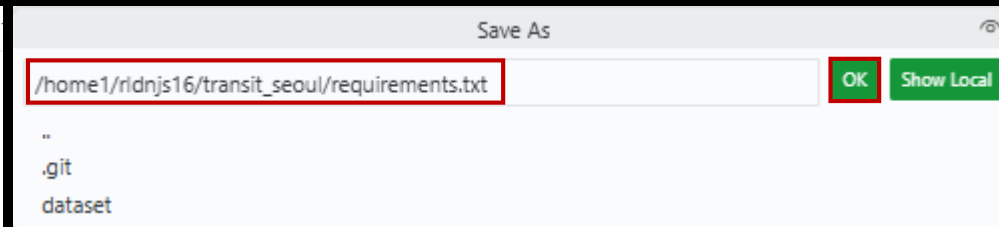
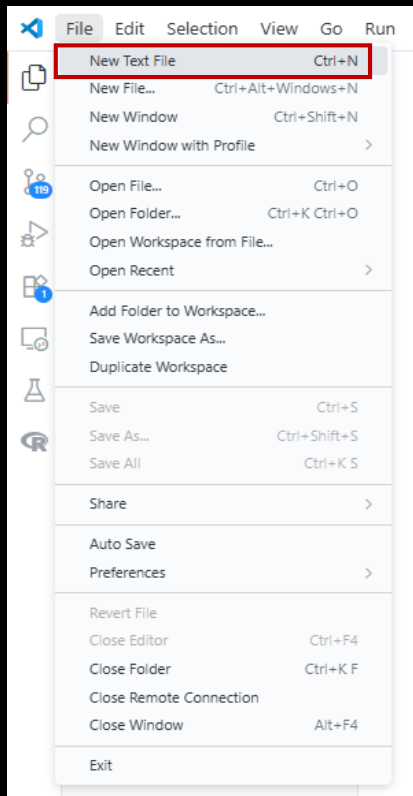


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● (base) [rldnjs16@gate2 transit_seoul]$ conda activate ubai  
○ (ubai) [rldnjs16@gate2 transit_seoul]$
```

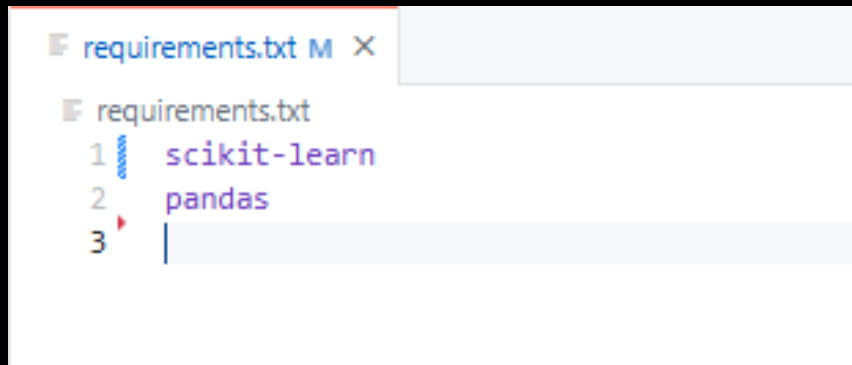
## Miniconda 실행

- requirements.txt
  - 앞서 Github에서 다운로드한 requirements.txt의 내용을 복사한 후,  
새 파일에 붙여넣고 requirements.txt라는 이름으로 저장해 주세요.

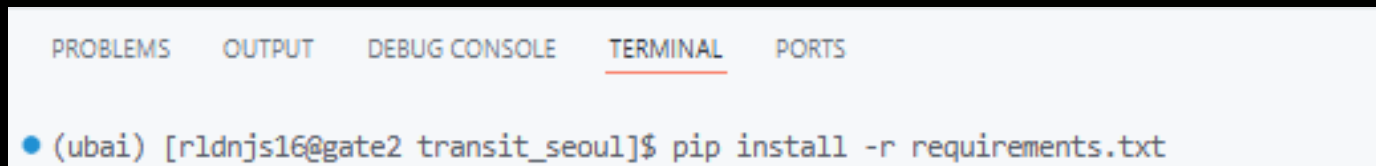


### Miniconda 실행

- requirements.txt
  - terminal에서 가상환경 접속 후 `pip install -r requirements.txt`를 입력합니다.
  - 설치하고자 하는 패키지들을 한 번에 설치합니다.
  - 이후 가상환경에서 별도 설치 없이 패키지를 사용할 수 있습니다.

A screenshot of a code editor window with a tab titled 'requirements.txt'. The editor shows the following content:

```
requirements.txt
1  scikit-learn
2  pandas
3  |
```

A screenshot of a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active and shows the command:

```
(ubai) [rldnjs16@gate2 transit_seoul]$ pip install -r requirements.txt
```



A nighttime photograph of a modern architectural complex. The main building features a large, curved facade made of dark, reflective panels, some of which are illuminated from within. In the foreground, a wide, paved plaza with a grid pattern is visible, with several people walking. In the background, other city buildings are lit up, and a street with a few vehicles is visible.

# Break Time

11시 10분까지 착석 부탁드립니다.

### Slurm

- 도시과학빅데이터 · AI연구원은 사용자에게 고성능컴퓨팅(HPC) 자원을 제공합니다.
- 사용자는 Slurm을 통해 독점적인 자원을 할당받고  
AI학습 및 추론, 연산, 시뮬레이션 등의 작업(Job)을 수행할 수 있습니다.
- Slurm은 다양한 사용자들의 다양한 요구를 수용하고(Job Submit),  
각 사용자들의 작업을 스케줄링하며(Task Scheduling),  
자원을 관리(Resource Management)하는 Linux 유틸리티 입니다.
- 여러 명의 사용자가 UBAI 클러스터를 이용하는데 있어서  
원활한 실험 및 계산을 위한 중재자 역할을 수행합니다.
- UBAI는 여러 명의 사용자에게 UBAI Cluster를 제공하기 위해 Slurm을 사용합니다.

# 04 Slurm 개념 및 Job 제출

## Cluster

- 여러 대의 컴퓨터들이 연결되어 하나의 시스템처럼 동작하는 컴퓨터들의 집합을 말합니다.

## Partition

- Partition은 특정 자원 그룹을 정의하는 논리적 단위입니다.
- 사용자들이 작업을 제출할 때 특정 Partition을 지정하여 자원을 할당받을 수 있습니다.

Partition	# of Nodes	# of Cores/ node	CPU	GPU/node	Memory/node	Scratch
gpu1	14(n001–n014)	48	Intel Xeon Gold 6240R	RTX3090(4EA)	768GB	500GB
gpu6	25(n015–n039)	48	Intel Xeon Gold 6240R	A10(4EA)	768GB	500GB
cpu1	10(n040–n049)	48	Intel Xeon Gold 6240R	–	768GB	500GB
gpu2	11(n051–n061)	56	Intel Xeon Gold 6348R	A10(4EA)	1024GB	500GB
gpu3	10(n062–n071)	56	Intel Xeon Gold 6348R	A6000ada(4EA)	1024GB	500GB
gpu4	29(n072–n100)	56	Intel Xeon Gold 6348R	A6000(4EA)	1024GB	500GB
gpu5	6(n101–n106)	64	Intel Xeon Platinum–8358	A6000(4EA)	1024GB	500GB

### Node

- 클러스터를 구성하는 개별 컴퓨터(서버)를 의미합니다.
- 각 Node는 CPU, 메모리, GPU, 디스크 등의 자원을 갖습니다.
- 대회 기간 동안 각 팀은 한 개의 노드를 독점적으로 사용합니다.

### Job

- 컴퓨터 클러스터와 같은 경우  
스케줄러(이하 Slurm)를 사용하여 컴퓨터 자원에 접근하기 위한 요청을 합니다.
- 이를 작업(Job)이라고 부르며,  
이는 사용자가 원하는 프로그램이나 스크립트를 실행하고 싶을 때의 행동을 지칭하기도 합니다.

## 배치 파일 생성

The screenshot shows the GitHub interface for the repository 'transit seoul'. At the top, there's a header with the repository name and a 'Public' badge. Below this, there are navigation elements like 'main' branch, '1 Branch', and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and 'Code' are also visible. The main content area displays a list of files and commits. The file 'run.sh' is highlighted with a red box. The commit history shows several commits, with the most recent one being 'e42d000' from user 'ki5n2'.

File	Commit Message	Time
README.md	Update README.md	5 hours ago
edu.txt	Add files via upload	8 minutes ago
requirements.txt	Add files via upload	13 minutes ago
run.sh	Create run.sh	5 hours ago
test.py	Add files via upload	7 minutes ago
transit_seoul_sample_code_2.py	Add files via upload	3 minutes ago
transit_seoul_sample_code_3.py	Add files via upload	3 minutes ago
해커톤_슈퍼컴퓨터 사용법 교육.pdf	Add files via upload	now

# 04 Slurm 개념 및 Job 제출

## 배치 파일 생성

- 배치 파일을 만들어주기 위해, 파일을 전부 복사해줍니다.

transit\_seoul / run.sh

ki5n2 Add files via upload
81eb4a7 · 2 minutes ago History

Code Blame 27 lines (22 loc) · 1.18 KB
Raw

```

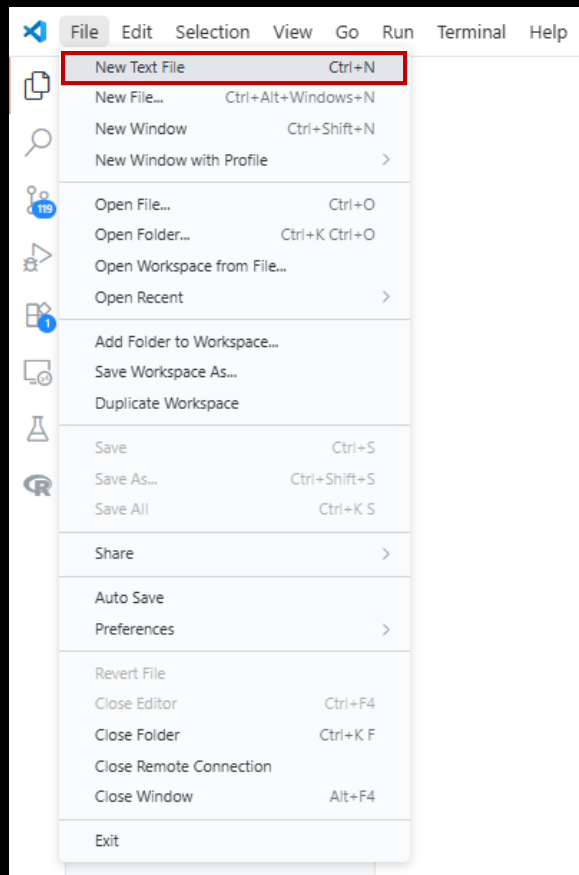
1  #!/bin/bash
2  #SBATCH --job-name=transit_seoul      # 작업 이름
3  #SBATCH --output=./output/%j.out
4  #SBATCH --error=./output/%j.err
5  #SBATCH --nodes=1
6  #SBATCH --partition=gpu6             # 사용할 파티션 이름
7  #SBATCH --nodelist=n036              # 사용할 노드 이름
8  #SBATCH --gres=gpu:1
9  #SBATCH --cpus-per-task=1            # 하나의 태스크가 사용할 CPU 코어 수
10 #SBATCH --mem=128G                   # 메모리 할당량
11 #SBATCH --time=48:00:00              # 최대 실행 시간 (형식: HH:MM:SS)
12
13 echo "start at: $(date)"              # 접속한 날짜 표기
14 echo "node: $HOSTNAME"               # 접속한 노드 번호 표기
15 echo "jobid: $SLURM_JOB_ID"          # jobid 표기
16
17 # Load modules (cuda 환경)
18 module load cuda/11.8.0
19
20 # Load env (python 환경)
21 source ~/miniconda3/etc/profile.d/conda.sh
22
23 # 가상환경 활성화
24 conda activate ubai                  # ubai라는 conda 환경에서, 슈퍼컴퓨팅 쓸 준비 완료
25
26 # python 스크립트 실행
27 python test.py

```

그림 이모티콘  
잘라내기  
복사  
붙여넣기  
일반 텍스트로 붙여넣기  
전체 선택  
NAVER에서 '#!/bin/bash #SBATCH --job-name=transit\_seoul...' 검색  
인쇄  
맞춤법 검사  
쓰기 방향  
음성으로 듣기  
쿼서치  
스크립북에 추가  
검사

## 배치 파일 생성

- 배치 파일 생성을 위해 새로운 텍스트 파일을 만들어줍니다.



# 04 Slurm 개념 및 Job 제출

## 배치 파일 생성

- 파일명 **run.sh**로 저장해줍니다.(예시, run.sh)

The screenshot shows a terminal window with a Slurm batch script and a 'Save As' dialog box. The terminal window has a menu bar with 'View', 'Go', 'Run', 'Terminal', and 'Help'. The script content is as follows:

```

$ #!/bin/bash Untitled-1
1  #!/bin/bash
2  #SBATCH --job-name=transit_seoul      # 작업 이름
3  #SBATCH --output=./output/%j.out
4  #SBATCH --error=./output/%j.err
5  #SBATCH --partition=gpu6              # 사용할 파티션 이름
6  #SBATCH --odelist=n036                # 사용할 노드 이름
7  #SBATCH --gres=gpu:1
8  #SBATCH --cpus-per-task=1             # 하나의 태스크가 사용할 CPU 코어 수
9  ##SBATCH --mem=128G                   # 메모리 할당량
10 ##SBATCH --time=48:00:00              # 최대 실행 시간 (형식: HH:MM:SS)
11
12 echo "start at: $(date)"              # 접속한 날짜 표기
13 echo "node: $HOSTNAME"                 # 접속한 노드 번호 표기
14 echo "jobid: $SLURM_JOB_ID"            # jobid 표기
15
16 # Load modules (cuda 환경)
17 module load cuda/11.8.0
18
19 # Load env (python 환경)
20 source ~/miniconda3/etc/profile.d/conda.sh
21
22 # 가상환경 활성화
23 conda activate ubai                    # ubai라는 conda 환경에서, 슈퍼컴퓨팅 쓸 준비 완료
24
25 # python 스크립트 실행
26 python test.py
  
```

The 'Save As' dialog box is open, showing the file path `/home1/ridnjs16/transit_seoul/run.sh` in the text field. Below the text field, there are two buttons: 'OK' and 'Show Local'. The file list below the buttons shows a single file named `dataset`.



# 04 Slurm 개념 및 Job 제출

## 배치 파일 생성

- 프로젝트 수행을 위한 배치 파일을 생성합니다.
- 다음의 사항을 본인의 작업(job)에 맞게 입력한 후, 본인이 원하는 파일의 이름을 지정하여 filename.sh 형식으로 파일을 저장합니다.
- 예를 들어, run.sh로 저장해보겠습니다.
- 파일명이 .sh 형식인지 반드시 확인해주세요.

```
$ run.sh U X
$ run.sh
1  #!/bin/bash
2  #SBATCH --job-name=transit_seoul          # 작업 이름
3  #SBATCH --output=./output/%j.out
4  #SBATCH --error=./output/%j.err
5  #SBATCH --partition=gpu6                  # 사용할 파티션 이름
6  #SBATCH --nodelist=n036                   # 사용할 노드 이름
7  #SBATCH --gres=gpu:1
8  #SBATCH --cpus-per-task=1                 # 하나의 태스크가 사용할 CPU 코어 수
9  #SBATCH --mem=128G                         # 메모리 할당량
10 #SBATCH --time=48:00:00                   # 최대 실행 시간 (형식: HH:MM:SS)
11
12 echo "start at: $(date)"                  # 접속한 날짜 표기
13 echo "node: $HOSTNAME"                   # 접속한 노드 번호 표기
14 echo "jobid: $SLURM_JOB_ID"              # jobid 표기
15
16 # Load modules (cuda 환경)
17 module load cuda/11.8.0
18
19 # Load env (python 환경)
20 source ~/miniconda3/etc/profile.d/conda.sh
21
22 # 가상환경 활성화
23 conda activate ubai                      # ubai라는 conda 환경에서, 슈퍼컴퓨팅 쓸 준비 완료
24
25 # python 스크립트 실행
26 python test.py
```

## 배치 파일 생성

#SBATCH --job-name=JOB\_NAME

- 작업 이름을 지정하는 명령어입니다.
- Ex) #SBATCH --job-name=transit\_seoul

#SBATCH --output=./output/%j.out

- 작업 결과 파일의 저장 위치를 지정하는 명령어입니다.
- Ex) #SBATCH --output=./output/%j.out

#SBATCH --error=./output/%j.err

- 작업 중 발생하는 오류 메시지의 파일 저장 위치를 지정하는 명령어입니다.
- Ex) #SBATCH --error=./output/%j.err

## 배치 파일 생성

```
#SBATCH --partition=gpu5
```

- 사용할 Partition을 지정하는 명령어입니다.
- 지정된 노드가 속한 파티션을 입력해줍니다.

```
#SBATCH --nodelist=n107
```

- 사용할 노드를 지정하는 명령어입니다.
- 각 팀 별로 지정된 노드를 입력해줍니다.

```
#SBATCH --gres=gpu:1
```

- 몇 개의 GPU를 사용할 것인지 지정하는 명령어입니다.
- 각 노드는 4개의 GPU를 가지고 있으며, 해당 옵션을 사용하면 해당 노드의 GPU 중 1개를 현재 작업에 할당함을 의미합니다.

## 배치 파일 생성

```
#SBATCH --cpus-per-task=1
```

- 총 필요한 CPU 코어의 개수를 지정하는 명령어입니다.

```
#SBATCH --mem=128G
```

- 작업을 실행할 때 필요한 메모리를 요청합니다.
- 요청한 메모리 이상을 초과해서 사용하려고 한다면, 해당 작업은 자동으로 중단됩니다.
- 따라서 메모리를 충분히 설정하거나, “`##SBATCH --mem=128G`” 을 통해 기본값(메모리 제한 없음)으로 실행합니다.
- 그럼에도, 각 노드가 갖는 메모리(1024G)를 초과하여 사용할 경우, 해당 작업은 중단됩니다.

```
#SBATCH --time=24:00:00
```

- 최대 실행 시간을 설정할 수 있습니다.
- 작업에 필요한 시간을 알기 어려우므로, “`##SBATCH --time=24:00:00`” 을 통해 기본값(48시간, 최대)으로 실행합니다.

## 배치 파일 생성

```
echo "start at: $(date)"
```

- 접속 날짜가 표기됩니다.

```
echo "node: $HOSTNAME"
```

- 접속한 노드 번호가 표기됩니다.

```
echo "jobid: $SLURM_JOB_ID"
```

- jobid가 표기됩니다.

```
module ~
```

- 원하는 Linux 환경을 구축할 수 있고, 기본적으로 CUDA/11.2.2 실행으로 셋팅되어 있습니다.
- 지금과 같이 다른 GPU 환경을 원할 경우, 해당 모듈을 unload한 후, 원하는 모듈을 load 합니다.
- GPU 환경을 사용하고 싶은 경우에만 해당하며, GPU 환경을 사용하지 않을 경우(CPU Partition 사용) 지우셔도 무관합니다.

### 배치 파일 생성

```
source ~/miniconda3/etc/profile.d/conda.sh
```

- Conda 환경을 사용하기 위한 초기 설정을 불러오는 작업입니다.
- 해당 명령을 통해 conda activate 등 Conda 관련 명령어들을 사용할 수 있도록 설정합니다.

```
conda activate ubai
```

- 본인이 생성한 가상환경을 활성화시켜,  
해당 환경에서 설치된 패키지들을 사용하여 작업을 진행할 수 있습니다.

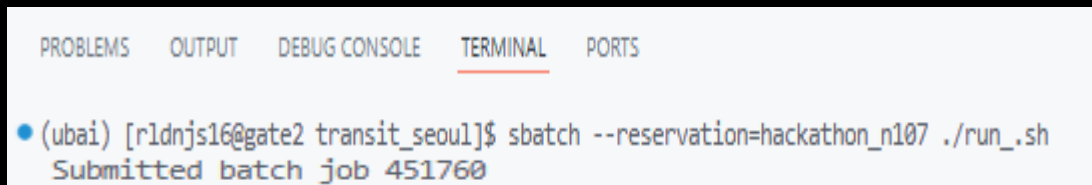
```
python test.py
```

- 원하는 Python 파일을 실행합니다.
- 실행하려는 파일은 .py 파일의 형태로 존재해야 합니다.

## 04 Slurm 개념 및 Job 제출

### 배치 파일 실행

- Terminal에 **sbatch** 명령어를 이용하여 **배치 파일명**을 입력 및 실행하세요.
- (★) 팀 별로 독점적으로 할당받은 노드를 사용하기 위해, 다음과 같이 배치 파일을 실행합니다.
  - **sbatch --reservation=hackathon\_팀별노드 ./filename.sh**
  - ‘reservation=hackathon\_팀별노드’ : 노드(팀 할당) 사용을 위한 예약 이름으로, 예약 이름으로만 해당 노드에 대한 작업 제출 가능합니다.
  - (예시) **sbatch --reservation=hackathon\_n107 ./run.sh**
- 이는 할당받은 노드에 작업(job)을 제출한다는 의미입니다.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

• (ubai) [r1dnjs16@gate2 transit_seoul]$ sbatch --reservation=hackathon_n107 ./run_.sh
Submitted batch job 451760
```

- 작업(job) 제출이 정상적으로 진행되었다면, **output** 폴더 안에 해당 작업에 대한 **err/out** 파일이 각각 생성됩니다.

## 04 Slurm 개념 및 Job 제출

### 배치 파일 실행

- 작업 상태 확인
  - 만일 해당 파일이 생성되지 않았다면, 해당 노드가 이미 제출한 다른 작업(job)으로 모두 할당되어 작업 대기 중일 가능성이 큽니다.
    - terminal에 `qstat` 명령어를 입력하여, 본인의 ID를 확인해 작업 현황을 점검하시기 바랍니다.
  - 배정이 되어 실행되고 있다면, Use 란에 **R**이라고 표시되어 있지만, 작업을 대기하고 있는 경우 **Q**라고 표시되어 있습니다.
  - 작업이 대기(**Q**)에 걸린 경우, 앞선 작업들이 끝나는 것을 기다려야 합니다.



- qstat: 모든 제출된 작업 상태를 확인
  - 사용시 특정 사용자가 제출한 작업만을 확인하기 위해서는 -u 옵션이 필요합니다.
  - qstat -u <user\_name>
  - Ex) qstat -u rldnjs16

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
• (ubai) [rldnjs16@gate2 transit_seoul]$ qstat -u rldnjs16

gate2.hpc:

Job id                Username Queue      Name                SessID NDS    TSK      Req'd  Req'd   Elap
-----  -
455426                rldnjs16 gpu6      transit_seoul      --      1      1        --  48:00 R  00:00

```

## 제출 작업 취소

- 제출한 작업을 취소하기 위해서는 scancel 명령어를 사용하세요.
  - scancel <job\_id> : 특정 작업만 취소합니다.
  - Ex) scancel 455426

```
(ubai) [rldnjs16@gate2 transit_seoul]$ scancel 455426
(ubai) [rldnjs16@gate2 transit_seoul]$ qstat -u rldnjs16

gate2.hpc:

Job id                Username Queue      Name                        SessID NDS   TSK   Req'd  Req'd   Elap
-----
455426                rldnjs16 gpu6      transit_seoul              --      1    1      --  48:00 C 00:00
```

- scancel -u <user\_name> : 자신이 제출한 모든 작업을 취소합니다.
  - Ex) scancel -u rldnjs16
- Use 란에 **C**가 나타났다면, 정상적으로 작업이 취소된 것입니다.

## Python 프로젝트 생성

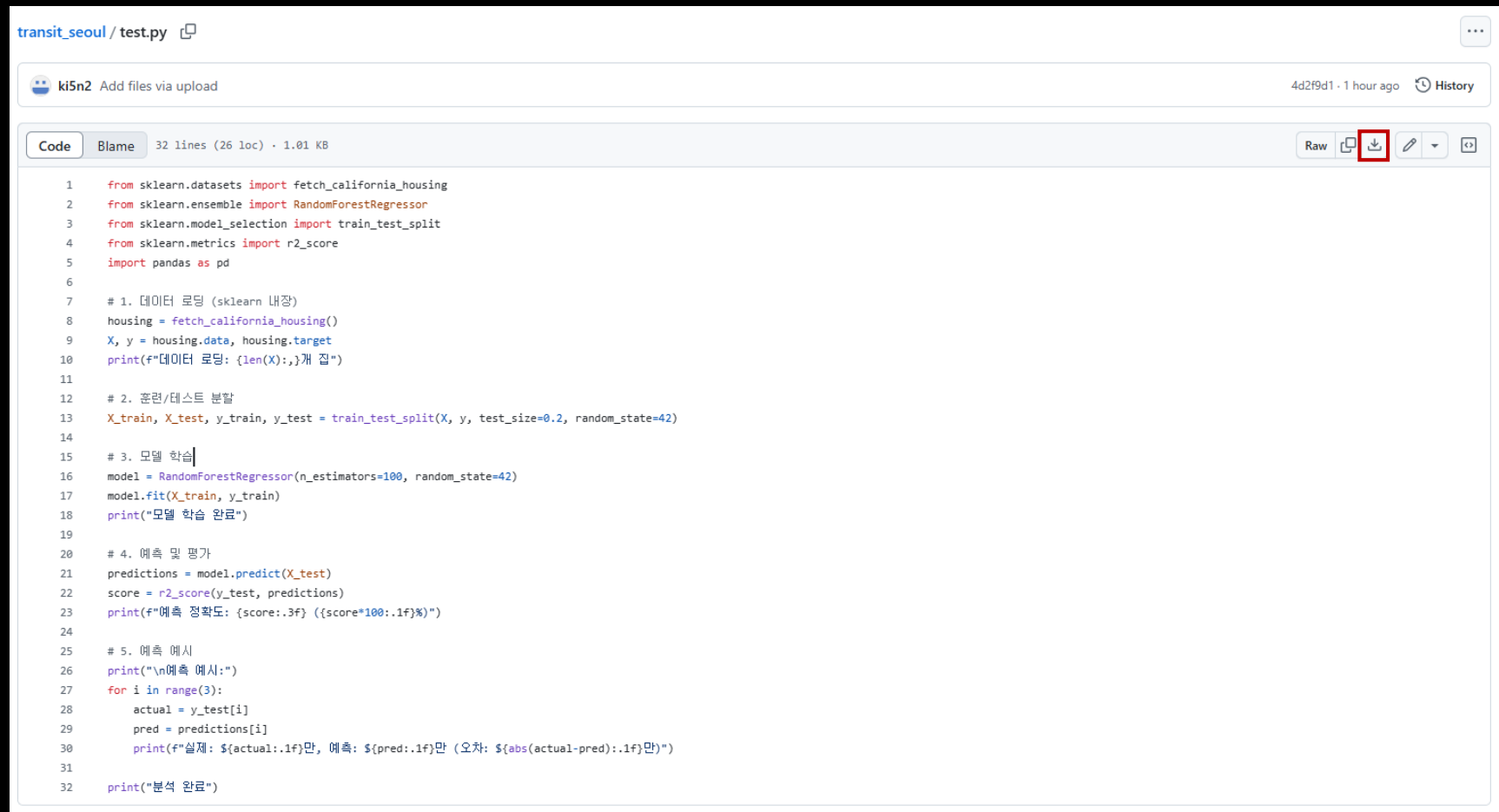
- 캘리포니아 집값 예측 테스트
  - [https://github.com/uos-ubai/transit\\_seoul/blob/main/test.py](https://github.com/uos-ubai/transit_seoul/blob/main/test.py)

The screenshot shows the GitHub repository page for 'transit\_seoul'. The repository is public and has 1 branch (main) and 0 tags. The file list is as follows:

File Name	Commit Message	Time Ago
README.md	Update README.md	5 hours ago
edu.txt	Add files via upload	8 minutes ago
requirements.txt	Add files via upload	13 minutes ago
run.sh	Create run.sh	5 hours ago
test.py	Add files via upload	7 minutes ago
transit_seoul_sample_code_2.py	Add files via upload	3 minutes ago
transit_seoul_sample_code_3.py	Add files via upload	3 minutes ago
해커톤_슈퍼컴퓨터 사용법 교육.pdf	Add files via upload	now

## Python 프로젝트 생성

- 캘리포니아 집값 예측 테스트
  - [https://github.com/uos-ubai/transit\\_seoul/blob/main/test.py](https://github.com/uos-ubai/transit_seoul/blob/main/test.py)



```
transit_seoul / test.py

ki5n2 Add files via upload 4d2f9d1 · 1 hour ago History

Code Blame 32 lines (26 loc) · 1.01 KB Raw Copy Download Edit History

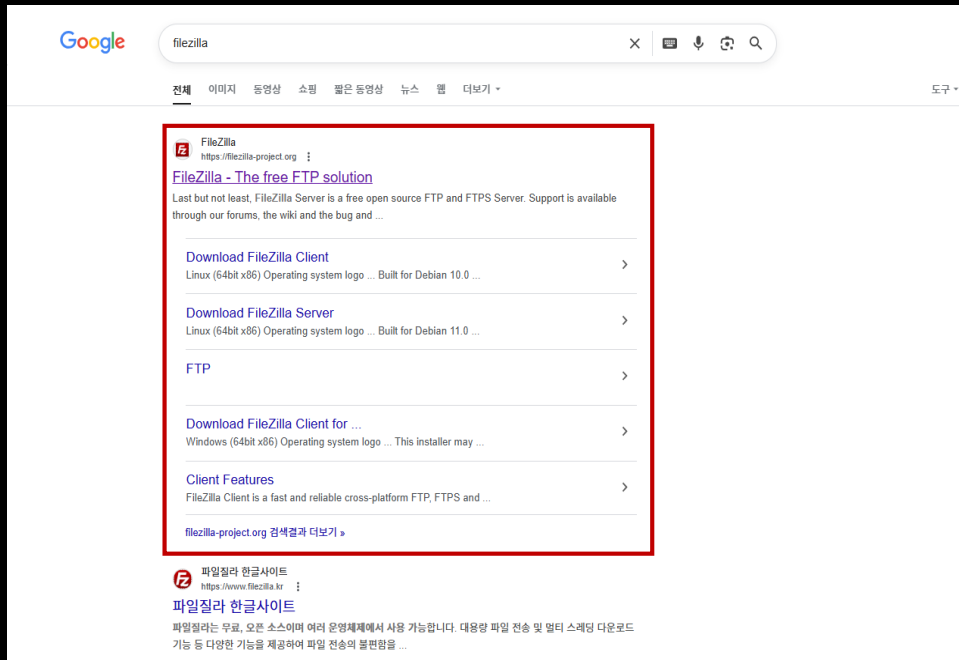
1 from sklearn.datasets import fetch_california_housing
2 from sklearn.ensemble import RandomForestRegressor
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import r2_score
5 import pandas as pd
6
7 # 1. 데이터 로딩 (sklearn 내장)
8 housing = fetch_california_housing()
9 X, y = housing.data, housing.target
10 print(f"데이터 로딩: {len(X):,}개 집")
11
12 # 2. 훈련/테스트 분할
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
14
15 # 3. 모델 학습
16 model = RandomForestRegressor(n_estimators=100, random_state=42)
17 model.fit(X_train, y_train)
18 print("모델 학습 완료")
19
20 # 4. 예측 및 평가
21 predictions = model.predict(X_test)
22 score = r2_score(y_test, predictions)
23 print(f"예측 정확도: {score:.3f} ({score*100:.1f}%)")
24
25 # 5. 예측 예시
26 print("\n예측 예시:")
27 for i in range(3):
28     actual = y_test[i]
29     pred = predictions[i]
30     print(f"실제: ${actual:.1f}만, 예측: ${pred:.1f}만 (오차: ${abs(actual-pred):.1f}만)")
31
32 print("분석 완료")
```

## 기타 프로그램

- 파일질라(FileZilla)
  - 파일 전송 프로그램
  - 로컬 컴퓨터와 원격 서버 간에 파일을 주고 받을 수 있음
  - 원격 서버에서 로컬 컴퓨터로 파일 다운로드 시,  
반드시(필수) 해당 프로그램(또는 WinSCP)을 통해 파일 전송

## 파일질라(FileZilla) 사용법

- 다운로드



A Google search result for 'filezilla'. The search bar shows 'filezilla' and the results list the FileZilla project page. A red box highlights the search results, which include the FileZilla logo, the URL 'https://filezilla-project.org', the title 'FileZilla - The free FTP solution', and a brief description: 'Last but not least, FileZilla Server is a free open source FTP and FTPS Server. Support is available through our forums, the wiki and the bug and ...'. Below the description are links for 'Download FileZilla Client' (Linux), 'Download FileZilla Server' (Linux), 'FTP', 'Download FileZilla Client for ...' (Windows), and 'Client Features'. At the bottom, there is a link to 'filezilla-project.org 검색결과 더보기'.



The FileZilla website homepage. The header features the FileZilla logo and the tagline 'The free FTP solution'. A navigation menu on the left includes links to Home, FileZilla, Features, Screenshots, Download (highlighted), Documentation, FileZilla Pro, RemoteDrive, FileZilla Server, Download, Community, Forum, Wiki, General, FAQ, Support, Contact, License, Privacy Policy, Trademark Policy, Development, Source code, Nightly builds, Translations, Version history, Changelog, and Issue tracker. Other projects listed are libfilezilla and Octochess. The main content area includes a 'Promotions' banner for FileZilla Pro, an 'Overview' section, a 'Quick download links' section with buttons for 'Download FileZilla Client' and 'Download FileZilla Server', and a 'News' section with recent updates and bugfixes.

## 파일질라(FileZilla) 사용법

- 다운로드

FileZilla The free FTP solution

Home

FileZilla

- Features
- Screenshots
- Download
- Documentation
- FileZilla Pro
- RemoteDrive

FileZilla Server

- Download

Community

- Forum
- Wiki

General

- FAQ
- Support
- Contact
- License
- Privacy Policy
- Trademark Policy

Development

- Source code
- Nightly builds
- Translations
- Version history
- Changelog
- Issue tracker

Other projects

- libfilezilla
- Octochess

Sponsors:

- RouteMe
- NDC

Promotion:

FileZilla® Pro The Best FTP Solution GET IN NOW >

### Download FileZilla Client for Windows (64bit x86)

The latest stable version of FileZilla Client is 3.69.2

Please select the file appropriate for your platform below.

Windows (64bit x86)

Download FileZilla Client

This installer may include bundled offers. Check below for more options.

The 64bit versions of Windows 10 and 11 are supported.

More download options

Other platforms:

Not what you are looking for?

Show additional download options

FileZilla The free FTP solution

Home

FileZilla

- Features
- Screenshots
- Download
- Documentation
- FileZilla Pro
- RemoteDrive

FileZilla Server

- Download

Community

- Forum
- Wiki

General

- FAQ
- Support
- Contact
- License
- Privacy Policy
- Trademark Policy

Development

- Source code
- Nightly builds
- Translations
- Version history
- Changelog
- Issue tracker

Other projects

- libfilezilla
- Octochess

Sponsors:

- RouteMe
- NDC

Promotion:

FileZilla® Pro The Best FTP Solution GET IN NOW >

### Download FileZilla Client for Windows (64bit x86)

The latest stable version of FileZilla Client is 3.69.2

Please select the file appropriate for your platform below.

Windows (64bit x86)

Download FileZilla Client

This installer may include bundled offers. Check below for more options.

The 64bit versions of Windows 10 and 11 are supported.

More download options

Other platforms:

Not what you are looking for?

Show additional download options

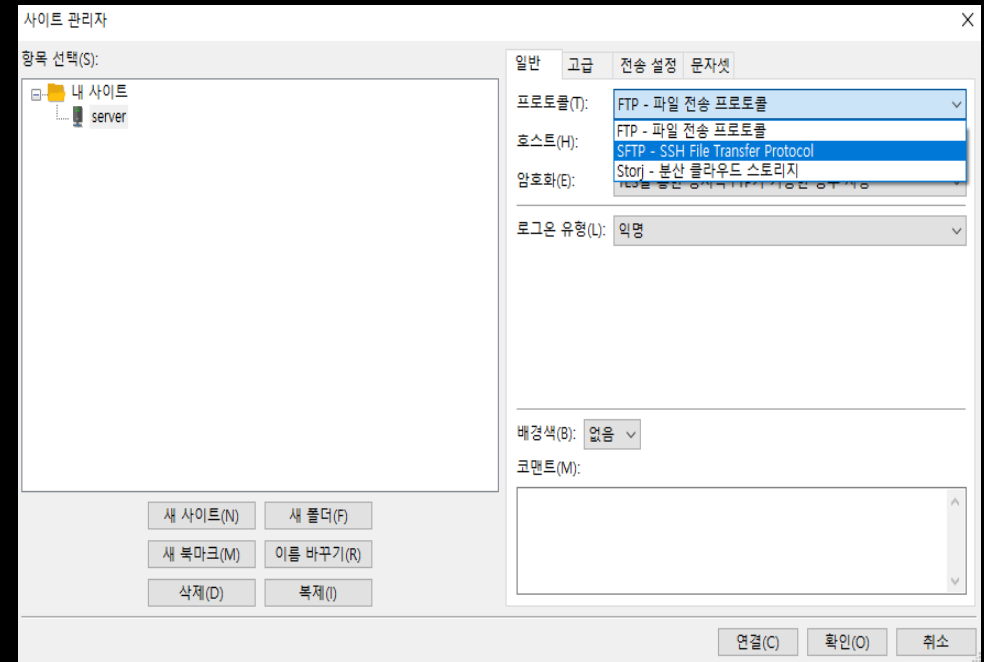
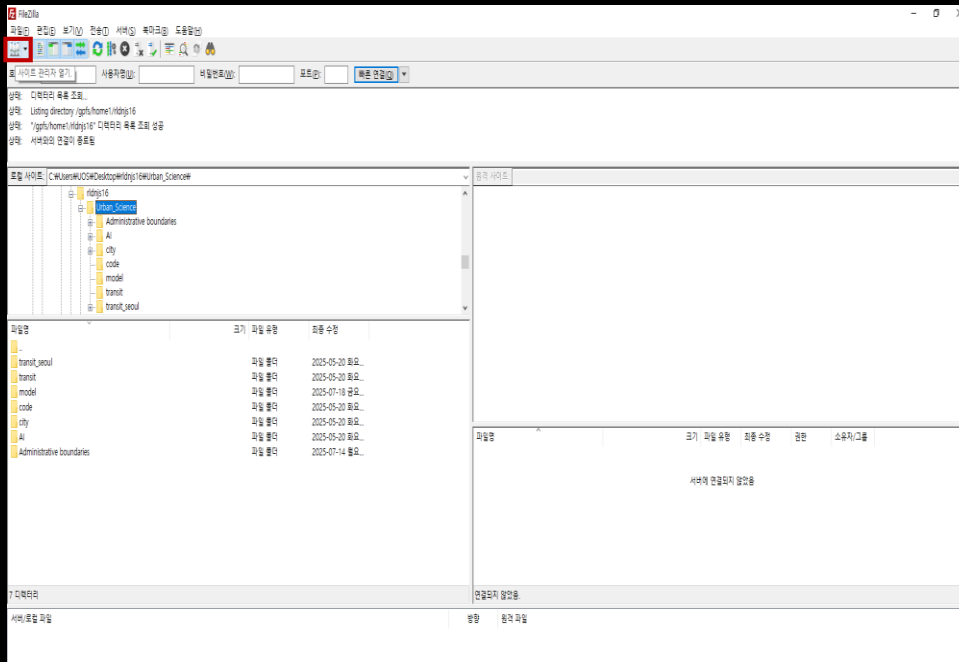
#### FileZilla Client: Select your edition

	FileZilla	FileZilla Bundle Includes FileZilla and FileZilla CLI	FileZilla Pro	FileZilla Pro Bundle Includes FileZilla Pro and FileZilla Pro CLI
FTP/FTPS/SFTP	✓	✓	✓	✓
Detailed Manual	X	✓	✓	✓
Batch Automation	X	✓	X	✓
Multi-cloud Support	X	X	✓	✓
Synchronization	X	X	✓	✓
	Download	Purchase	Purchase	Purchase

# 04 Slurm 개념 및 Job 제출

## 파일질라(FileZilla) 사용법

- 로컬 컴퓨터와 원격 서버 연결



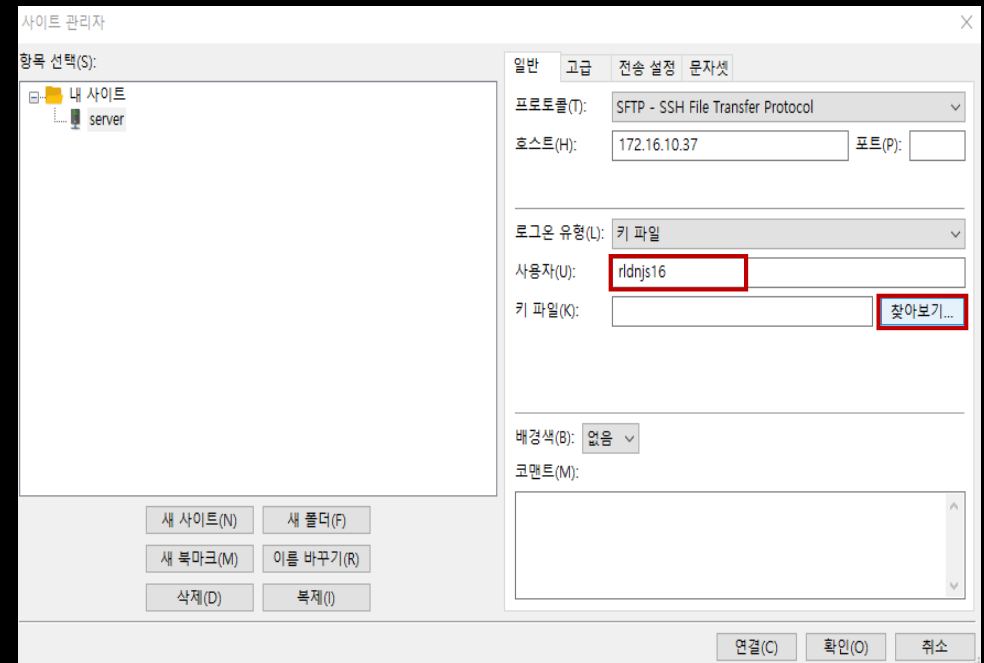
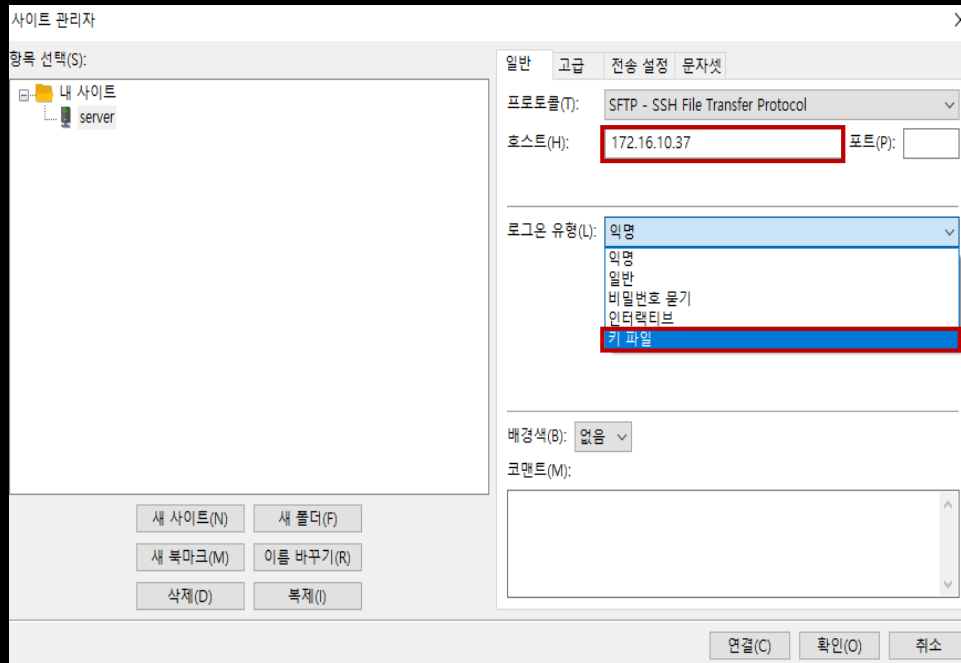
- 프로토콜을 **SFTP – SSH File Transfer Protocol** 로 변경해줍니다.



# 04 Slurm 개념 및 Job 제출

## 파일질라(FileZilla) 사용법

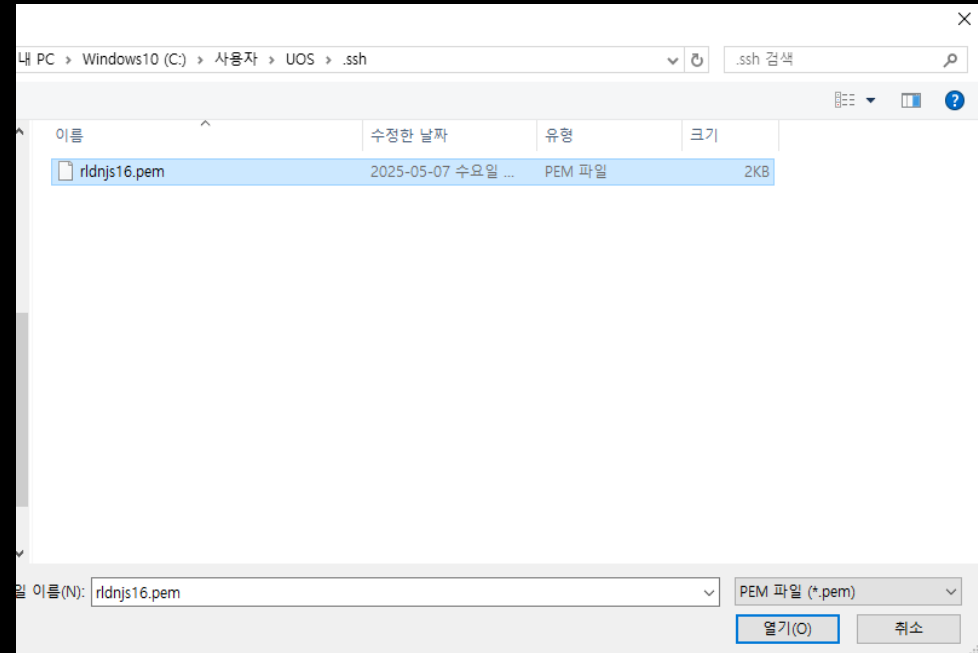
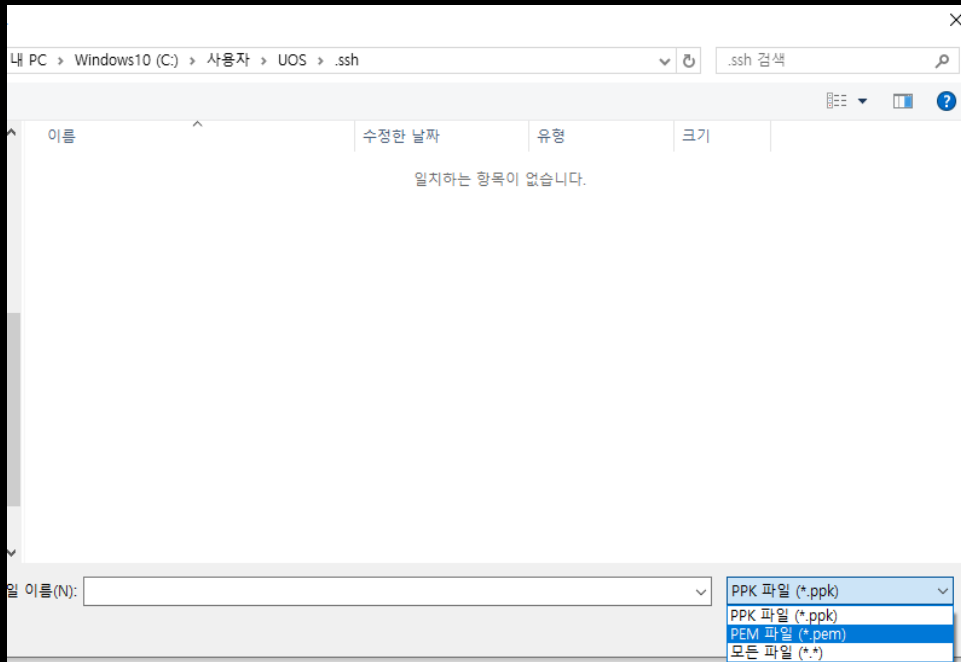
- 로컬 컴퓨터와 원격 서버 연결



- 호스트를 172.16.10.37로 입력해줍니다.
- 로그온 유형을 키 파일로 변경해줍니다.
- 사용자에 팀 계정명을 입력해줍니다.
- 키 파일 경로를 지정해줍니다.

## 파일질라(FileZilla) 사용법

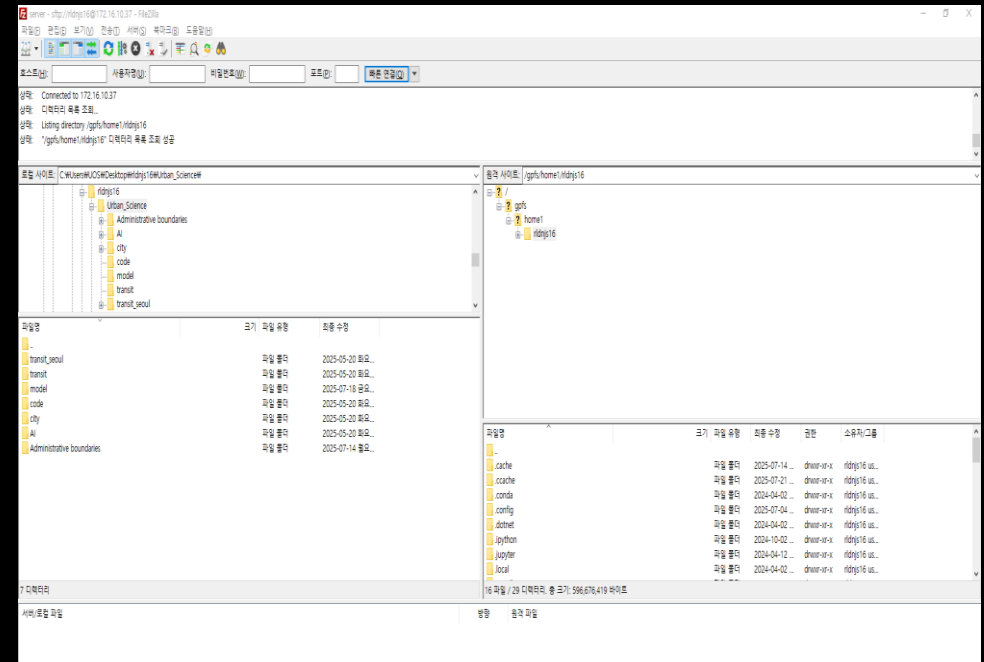
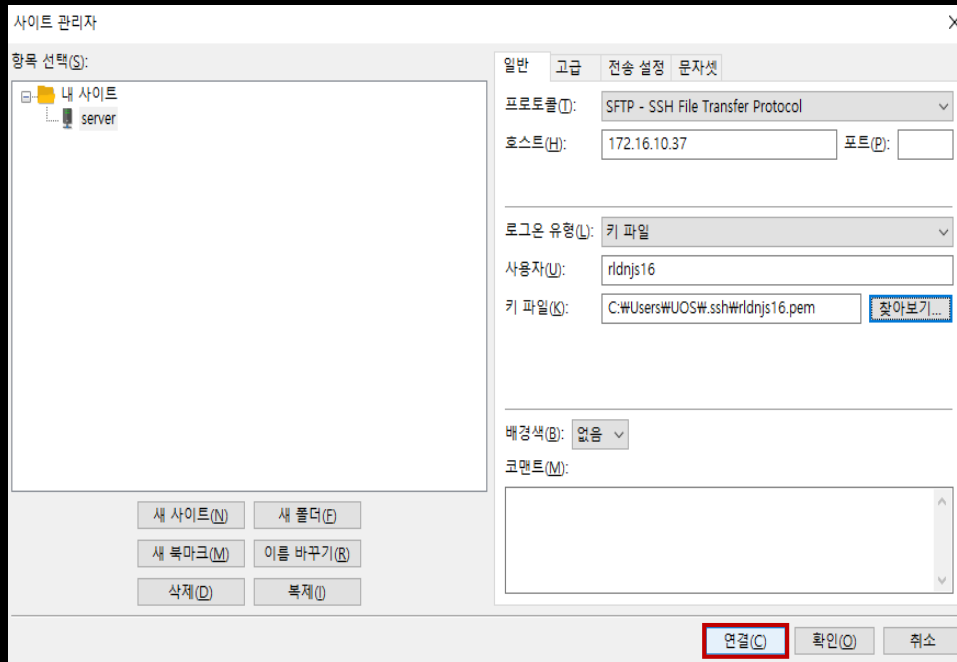
- 로컬 컴퓨터와 원격 서버 연결



- PPK 파일을 **PEM 파일**로 변경해줍니다.
- 발급받은 키 파일**을 선택해줍니다.

## 파일질라(FileZilla) 사용법

- 로컬 컴퓨터와 원격 서버 연결

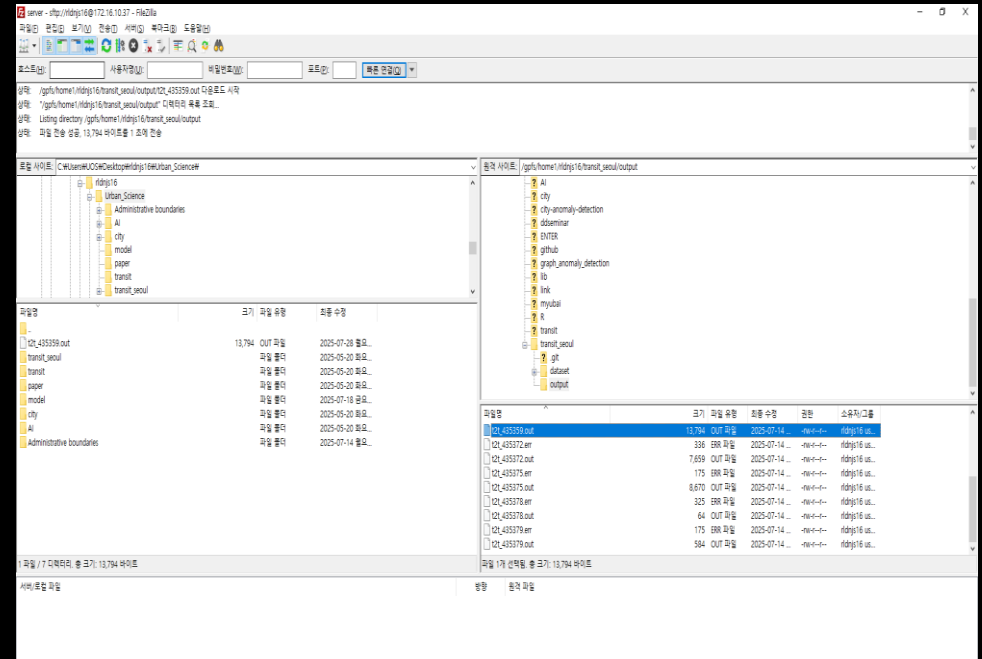
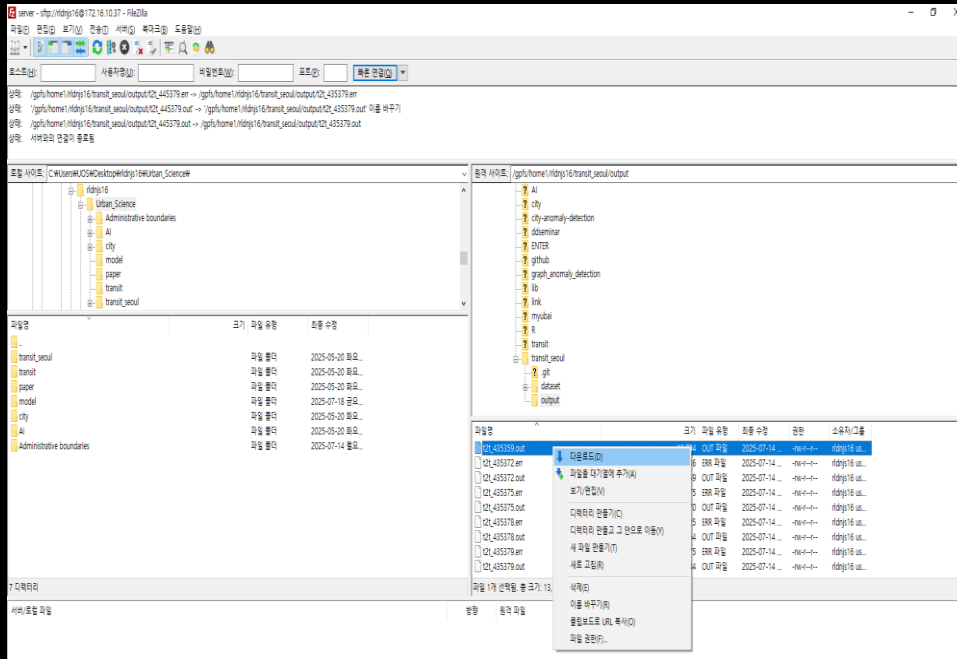


- 서버 내 파일들이 보인다면, 정상적으로 연결된 것입니다.

# 04 Slurm 개념 및 Job 제출

## 파일질라(FileZilla) 사용법

- 파일 전송



## 사사의 글 안내

도시과학빅데이터AI연구원의 슈퍼컴퓨터 자원을 이용하신 분들은  
논문, 프로젝트, 통계 등 실적에 다음과 같이 연구원 사사를 적어주시기 바랍니다.

(국문) 본 논문은 서울시립대학교 도시과학빅데이터AI연구원의 슈퍼컴퓨팅 자원을  
지원 받아 수행되었습니다.

(영문) The authors acknowledge the Urban Big data and AI Institute of the  
University of Seoul supercomputing resources (<http://ubai.uos.ac.kr>) made  
available for conducting the research reported in this paper.

**UBAI**





# Thank You

UBAI  
[UBAI@uos.ac.kr](mailto:UBAI@uos.ac.kr)