

# UBAI GUIDE BOOK

hs.hwang

2024-10-16

# Table of contents

<b>1</b>	<b>MAIN</b>	<b>4</b>
<b>2</b>	<b>UBAI Cluster</b>	<b>5</b>
	UBAI . . . . .	5
	Partition . . . . .	5
	. . . . .	7
<b>3</b>	<b>Chapter1.</b>	<b>8</b>
	1. . . . .	8
	2. . . . .	8
<b>4</b>	<b>Chapter2.</b>	<b>9</b>
	1. Remote-SSH . . . . .	9
	2. Config . . . . .	10
	3. SSH . . . . .	11
<b>5</b>	<b>Chapter3.</b>	<b>13</b>
	1. Linux . . . . .	13
	1.1 Module avail . . . . .	13
	1.2 Module show . . . . .	14
	1.3 Module load . . . . .	15
	1.4 Module list . . . . .	15
	1.5 Module rm . . . . .	15
	1.6 Module purge . . . . .	15
	2. Python . . . . .	16
	2.1 Minicoda . . . . .	16
	2.2 Minicoda . . . . .	17
	3. Enroot . . . . .	18
	5.0.1 3.1 Enroot . . . . .	18
	5.0.2 3.2 Image Container . . . . .	19
	5.0.3 3.3 Enroot . . . . .	19
<b>6</b>	<b>Chapter4. Python</b>	<b>23</b>
	1. BASH . . . . .	23
	1.1 . . . . .	23
	1.2 . . . . .	25

2. Jupyter Notebook	26
2.1	26
2.2	27
2.3	28
2.4 Jupyter Notebook	29
2.5 Jupyter Notebook	30
<b>7 Chapter5.</b>	<b>32</b>
1.	32
7.0.1 1.1	32
7.0.2 1.2	33
2.	33
7.0.1 2.1	33
7.0.2 2.2	38

# 1 MAIN

UBAI Cluster .

Chapter .

**UBAI Cluster**

UBAI Cluster .

**Chapter1.**

UBAI .

**Chapter2.**

UBAI .

**Chapter3.**

, .

**Chpater4. Python**

Python .

**Chpater5.**

, .

UBAI GUIDE BOOK UBAI .

· AI .



**서울시립대학교 | 도시과학빅데이터·AI연구원**

## 2 UBAI Cluster

### UBAI

· AI ( UBAI) (HPC) .  
Slurm AI , , (Job) .  
**Slurm**

Slurm (Job Submit),  
(Task Scheduling),  
(Resource Management) Linux .

UBAI UBAI Cluster Slurm .  
Slurm [Slurm](#) .

#### Visual Studio Code

Visual Studio Code( VScode) Microsoft .  
MacOS, Linux, Windows ,  
.

UBAI VScode .  
VScode . VScode , VScode .

### Partition

Slurm Partition .  
Partition .  
Partition .

Partition	# of Nodes	# of Cores/node	CPU	GPU/node	Memory/node	SSD	Note
gpu1	13	48	Intel Xeon Gold 6240R	RTX3090 (4EA)	768GB	2TB	*
edu1	5	48	Intel Xeon Gold 6240R	A10 (4EA)	768GB	2TB	*
cpu1	30	48	Intel Xeon Gold 6240R	None	768GB	2TB	*
gpu2	10	56	Intel Xeon Gold 6348R	A10 (8EA)	1024GB	2TB	*
gpu3	11	56	Intel Xeon Gold 6348R	A10 (4EA)	1024GB	2TB	*
gpu4	29	56	Intel Xeon Gold 6348R	A6000 (4EA)	1024GB	2TB	*
gpu5	6	64	Intel Xeon Platinum-8358	A6000 (4EA)	1024GB	2TB	*

※ UBAI 106 , 5,586 CPU , RTX3090 52 , A10 144 , A6000 140 .

Terminal Partition .

```
sinfo -o "%10P %5D %14F %4c %14G %N"
```

PARTITION	NODES	NODES(A/I/O/T)	CPUS	GRES	NODELIST
gpu1	13	10/3/0/13	48	gpu:rtx3090:4	n[001-013]
cpu1	35	16/19/0/35	48	(null)	n[014-048]
hgx	1	0/0/1/1	48	gpu:hgx:8	n050
gpu2	32	26/6/0/32	56	gpu:a10:4	n[051-070,073-080,083-086]
cpu2	14	14/0/0/14	56	(null)	n[087-100]
cpu3	6	4/2/0/6	64	(null)	n[101-106]
test	4	0/4/0/4	56	gpu:a10:4	n[071-072,081-082]

MaxJobs( ) 10, MaxSubmit( ) 20, MaxWall( ) 2 .

Partition	MaxJobs	MaxSubmit	MaxWall
*	10	20	2-00:00:00

AI , , .  
( ) AI .  
( ) 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.

### 3 Chapter1.

UBAI Cluster

1.

UBAI

(ubaisysadmin@uos.ac.kr)

2.

( ID.pem) C:\User\{ }\.ssh\

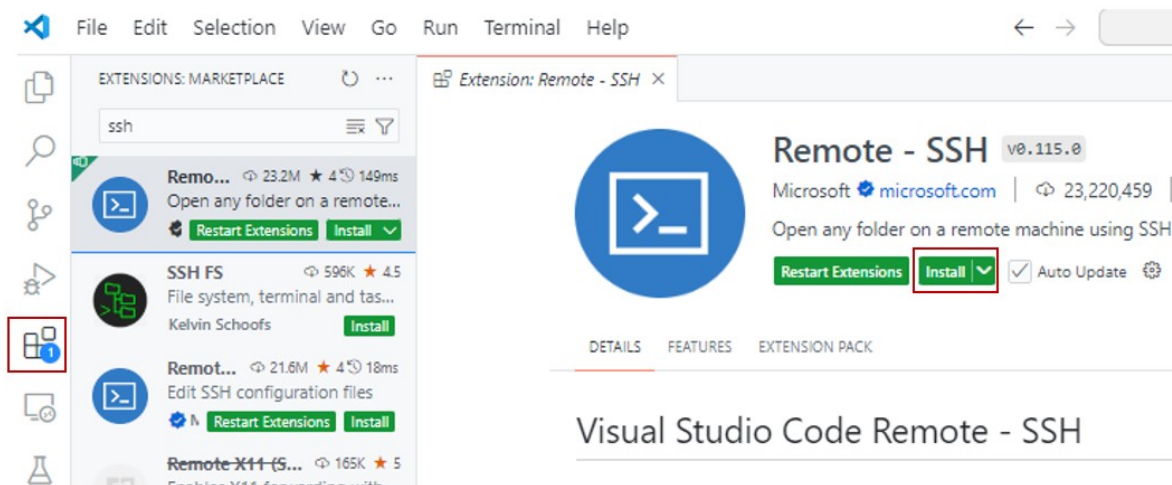
내 PC > Windows10 (C:) > 사용자 > UOS > .ssh			
이름	수정한 날짜	유형	크기
known_hosts	2024-10-10 목요일 ...	파일	1KB
known_hosts.old	2024-10-10 목요일 ...	OLD 파일	1KB
config	2024-10-10 목요일 ...	파일	1KB
ssu.pem	2024-10-07 월요일 ...	PEM 파일	3KB

※ .ssh , Chapter2. .ssh



## 4 Chapter2.

### 1. Remote-SSH



VScode Remote-SSH . SSH .  
SSH

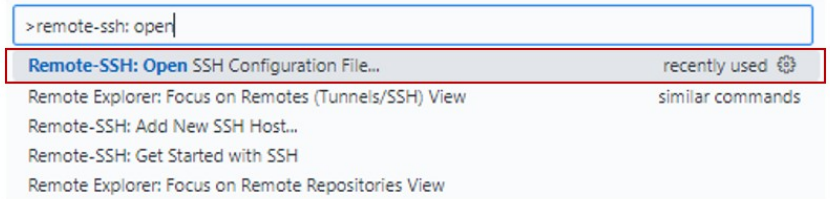
SSH  
command .

SSH , VScode .

VScode Extension Remote-SSH .

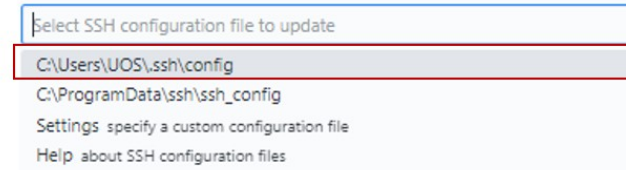
## 2. Config

1. Remote-SSH , (CTRL + P) (search) >remote-ssh : open ssh



configuration .

※ .ssh , .ssh config . .ssh config .



2. , C:\Users\ \.ssh\config .

3. .ssh config .

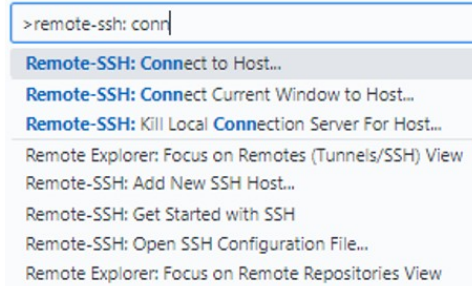
config

```
Host gate1
  HostName 172.16.10.36
  Port 22
  User ID
  IdentityFile

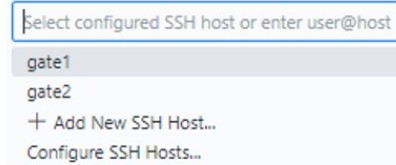
Host gate2
  HostName 172.16.10.37
  Port 22
  User ID
  IdentityFile
```

### 3. SSH

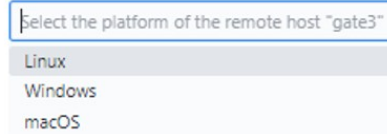
1. VScode (CTRL+SHIFT+P) .



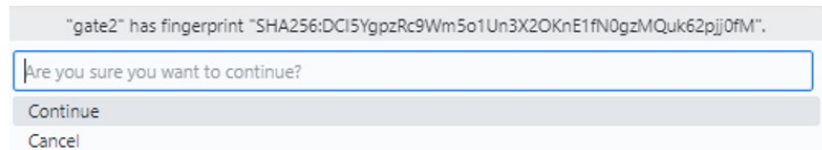
2. > remote-ssh : connect to host , .



3. gate1 gate2 . gate1 gate2 .



4. Linux .

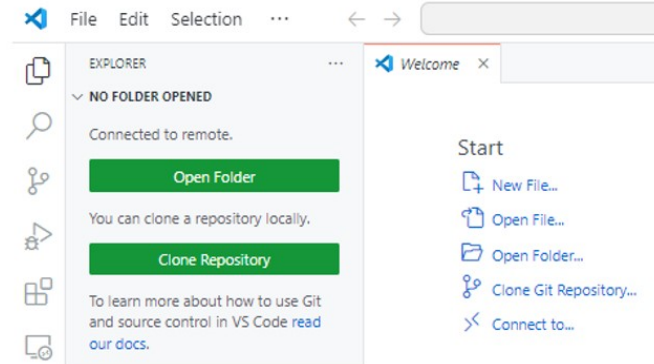


5. Continue .



SSH: gate1 0 0 0

6. gate , gate SSH:gate .



7. , (Explorer) , Open Folder .

8. /home1/{ ID} , OK .

9. , SSH !

## 5 Chapter3.

(OS), , .

OS, , , .

Linux Conda Enroot .

### 1. Linux

#### Environment Modules

Environment Modules

Unix/Linux

Environment Modules (PATH )

(<https://modules.sourceforge.net/>) .

#### 1.1 Module avail

UBIA Cluster .

```
(ubai) [ssu@gate1 ~]$ module avail
```

```
----- /opt/ohpc/pub/modulefiles -----
  CUDA/11.2.2      cmake/3.24.2      cuda/11.2.2
  EasyBuild/4.9.1  compiler-rt/latest  cuda/11.3.1
  R/4.3.1          compiler-rt/2023.1.0 (D)  cuda/11.4.4
```

advisor/latest	compiler-rt32/latest	cuda/11.5.2
advisor/2023.1.0 (D)	compiler-rt32/2023.1.0 (D)	cuda/11.6.2
autotools	compiler/latest	cuda/11.7.1
ccl/latest	compiler/2023.1.0 (D)	cuda/11.8.0
ccl/2021.9.0 (D)	compiler32/latest	cuda/12.0.0
clck/latest	compiler32/2023.1.0 (D)	cuda/12.1.1
clck/2021.7.3 (D)	cuda/leejihun_cuda	cuda/12.2.1 (D)

Where:

D: Default Module

If the avail list is too long consider trying:

"module --default avail" or "ml -d av" to just list the default modules.

"module overview" or "ml ov" to display the number of modules for each name.

Use "module spider" to find all possible modules and extensions.

Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "k

## 1.2 Module show

```
(ubai) [ssu@gate1 ~]$ module show cuda/11.2.2
```

```
-----
/opt/ohpc/pub/modulefiles/cuda/11.2.2:
-----
```

```
whatis("Name: CUDA Collection")
whatis("Version: 11.2.2")
whatis("Category: cuda")
prepend_path("PATH", "/opt/ohpc/pub/cuda/11.2.2/bin")
prepend_path("INCLUDE", "/opt/ohpc/pub/cuda/11.2.2/include")
prepend_path("LD_LIBRARY_PATH", "/opt/ohpc/pub/cuda/11.2.2/lib64")
family("cuda")
help([[
This module loads the CUDA

Version 11.2.2

]])
```

### 1.3 Module load

```
(ubai) [ssu@gate1 ~]$ module load cuda/11.2.2
```

### 1.4 Module list

```
(ubai) [ssu@gate1 ~]$ module list
```

Currently Loaded Modules:

```
1) cuda/11.2.2  2) dal/latest
```

### 1.5 Module rm

```
(ubai) [ssu@gate1 ~]$ module rm dal/latest
```

Removing dal version 2023.1.0

Use `module list` to view any remaining dependent modules.

### 1.6 Module purge

```
module rm          . module purge  module list  module
```

```
(ubai) [ssu@gate1 ~]$ module purge
```

```
(ubai) [ssu@gate1 ~]$ module list
```

No modules loaded

## 2. Python

### Python

Python , , ,  
, .  
, .

Anaconda Miniconda . UBAI Miniconda  
.

### Miniconda

Anaconda

.  
.

Miniconda Anaconda .

### 2.1 Minicoda

Miniconda Miniconda .

1. terminal terminal . Miniconda .  
`wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`
2. `wget` `bash` .  
`bash Miniconda3-latest-Linux-x86_64.sh`
3. Miniconda . Enter .  
'yes' . Enter .
4. Enter .
5. , `conda init` . 'yes' enter .
6. .
7. , terminal (base)[ ID@ \_gate\_number] . (Explorer)  
miniconda .



## 2.2 Minicoda

Miniconda , Python .

Python (Package Dependencies) . Python

1. terminal .

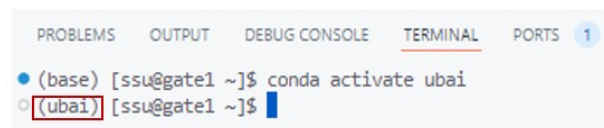
```
conda create -n { _ } python={ _Python_ }
```

ex. conda create -n ubai python=3.11

2. .

```
conda activate { _ }
```

ex. conda activate ubai



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1
• (base) [ssu@gate1 ~]$ conda activate ubai
○ (ubai) [ssu@gate1 ~]$
```

, conda activate .  
conda info --envs , .

Jupyter notebook , . pip install ipykernel jupyterlab or conda  
install ipykernel jupyterlab

※ Python *pip install* .

### 3. Enroot

```
enroot evidia , .
, ' ' , root .
NVIDIA enroot .
UBAI Rocky Linux 8.8 .
.
, /home /home root .
, sudo .
, Rocky Linux dnf , .
dnf search dnf ,
.
, Rocky Linux .
, ubuntu centos .
ubuntu OS .
```

#### 5.0.1 3.1 Enroot

```
enroot .
root .
2TB , enroot ( /enroot ).
enroot , srun !
( gate1 gate2 .)
gate .
srun .
enroot docker .
Rocky linux 8.8 , sudo sudo .
```

## 5.0.2 3.2 Image Container

enroot docker image container .  
docker , enroot docker , docker image container .  
image .  
 , .  
 .  
 , ubuntu ,  
 , ubuntu .  
image dockerhub .  
container .  
 .  
 , root .

## 5.0.3 3.3 Enroot

1. sbatch enroot slurm .  
 , Linux .

```
# gpu5  
srun --pty -p gpu5 -c 2 /bin/bash
```

```
○ s(base) [ssu@gate1 ~]$ srun --pty -p gpu5 -c 2 /bin/bash  
srun: job 263557 queued and waiting for resources  
srun: job 263557 has been allocated resources
```

## 2. Enroot

```
enroot (1~3), docker hub .  
docker hub docker:// .
```

```
# dockerhub  
enroot import docker://eclipse/ubuntu_python
```

```
(base) [ssu@n101 ~]$ enroot import docker://eclipse/ubuntu_python  
[INFO] Querying registry for permission grant  
[INFO] Authenticating with user: <anonymous>  
[INFO] Authentication succeeded  
[INFO] Fetching image manifest list  
[INFO] Fetching image manifest  
[INFO] Downloading 16 missing layers...  
  
100% 16:0=0s b234f539f7a1d65eabae1617e63c81ac01768abffd48b5cbbf7166  
[INFO] Extracting image layers...  
  
100% 15:0=0s b234f539f7a1d65eabae1617e63c81ac01768abffd48b5cbbf7166  
[INFO] Converting whiteouts...  
  
100% 15:0=0s b234f539f7a1d65eabae1617e63c81ac01768abffd48b5cbbf7166
```

## 3. Enroot

```
enroot .  
enroot .  
    , ubuntu.sqsh .  
    .
```

```
# ubuntu.sqsh  
enroot create -n mycontainer eclipse_ubuntu_python.sqsh
```

```
(base) [ssu@n101 ~]$ enroot create -n mycontainer eclipse+ubuntu_python.sqsh
[INFO] Extracting squashfs filesystem...

Parallel unsquashfs: Using 2 processors
32434 inodes (38725 blocks) to write

[=====]

created 27953 files
created 3229 directories
created 4461 symlinks
created 0 devices
created 0 fifos
```

#### 4. Enroot

```
# (version.1)
enroot start mycontainer
# (version.2)
enroot start --root --rw --mount ./mnt ubuntu-test /bin/bash
```

```
    |--mount ./mnt`
mnt      home      .
```

#### 5. Enroot

```
enroot      ,      .
enroot      .
      .
      ,      .
```

```
enroot export mycontainer new_image.sqsh
```

```
,      .
      ,      .
```

#### 6. Enroot

```
exit
```

```
`exit`    `Ctrl+D`    .
```

## 6 Chapter4. Python

Python .  
Shell Python . Shell Jupyter Notebook , Python .

### 1. BASH

Bash .  
(job) , (ubaisysadmin@uos.ac.kr)  
.  
.

#### 1.1

.  
(job) , filename.sh . Shell python\_project.sh  
.  
.sh .

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --partition=gpu2
#SBATCH --cpus-per-task=56
#SBATCH --gres=gpu:4
#SBATCH --job-name=UBAIJOB
#SBATCH -o ./ /jupyter.%N.%j.out # STDOUT
#SBATCH -e ./ /jupyter.%N.%j.err # STDERR

echo "start at:" `date`
```

```

echo "node: $HOSTNAME"
echo "jobid: $SLURM_JOB_ID"

module unload CUDA/11.2.2
module load cuda/11.8.0

python cnn.py 12 256 'relu'

```

STDOUT , STDERR (directory)

```

#SBATCH --nodes=1 , . nodes=1 .
#SBATCH --partition=gpu4 Partition . Partition UBAI Cluster .
#SBATCH --cpus-per-task=14 . n , 1 CPU/GPU
. #of Cores/node Partition . UBAI Cluster .
#SBATCH --gres=gpu:1 GPU . CPU Partition . GPU
, .
#SBATCH --job-name=UBAIJOB .
echo "start at:" 'date' .
echo "node: $HOSTNAME" .
echo "jobid: $SLURM_JOB_ID" jobid .
module ~ Linux . GPU , GPU (CPU Partition ) .
Chapter3. module envrionment .
python cnn.py 12 256 'relu' Python . .py . cnn.py
. Python sys sys.argv . sys .
python {filename}.py .

```



## 1.2

, Python .  
 , terminal sbatch . (job) .  
 (job) ID .

```
sbatch filename.sh # ex) sbatch python_project.sh
```

※ *cnn.py* *pip install tensorflow* ~~ex~~ *pip install numpy* .  
 (job) , STDOUT OUT .  
 OUT , Partition (job) . terminal *squeue* ,  
 ID .  
 n001, n002 ... , ( *Resources, Priority* ) .  
 , .  
 Partition Partition cpus-per-task, gpu Partition (job) .  
 STDOUT OUT .

```
jupytercn013.206248.out x
cn013_out > jupytercn013.206248.out
1 start at: Tue Oct 8 09:47:10 KST 2024
2 node: n013
3 jobid: 206248
4
5 ----- /opt/ohpc/pub/modulefiles -----
6   CUDA/11.2.2      cuda/11.3.1      cuda/11.6.2      cuda/12.0.0
7   cuda/leejihun_cuda  cuda/11.4.4      cuda/11.7.1      cuda/12.1.1
8   cuda/11.2.2      cuda/11.5.2      cuda/11.8.0      cuda/12.2.1 (D)
9
10  Where:
11  D: Default Module
12
13  If the avail list is too long consider trying:
14
15  "module --default avail" or "ml -d av" to just list the default modules.
16  "module overview" or "ml ov" to display the number of modules for each name.
17
18  Use "module spider" to find all possible modules and extensions.
19  Use "module keyword key1 key2 ..." to search for all possible modules matching
20  any of the "keys".
21
22  Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
23
24
25   8192/11490434 [.....] - ETA: 0s
26   16384/11490434 [.....] - ETA: 48s
27   40960/11490434 [.....] - ETA: 33s
```

## 2. Jupyter Notebook

Jupyter notebook

(job) , (ubaisysadmin@uos.ac.kr)

### 2.1

(job) , filename.sh . Shell  
jupyter\_notebook.sh .  
.sh .

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --partition=gpu4
#SBATCH --cpus-per-task=14
#SBATCH --gres=gpu:1
#SBATCH --job-name=UBAIJOB
#SBATCH -o ./ /jupyter.%N.%j.out # STDOUT
#SBATCH -e ./ /jupyter.%N.%j.err # STDERR

echo "start at:" `date`
echo "node: $HOSTNAME"
echo "jobid: $SLURM_JOB_ID"

module unload CUDA/11.2.2
module load cuda/11.8.0

python -m jupyter lab $HOME \
    --ip=0.0.0.0 \
    --no-browser
```

STDOUT , STDERR . (directory)

```

#SBATCH --nodes=1 , . nodes=1 .
#SBATCH --partition=gpu4 Partition . Partition UBAI Cluster .
#SBATCH --cpus-per-task=14 . n , 1 CPU/GPU
. #of Cores/node Partition . UBAI Cluster .
#SBATCH --gres=gpu:1 GPU . CPU Partition . GPU
, .
#SBATCH --job-name=UBAIJOB .
echo "start at:" 'date' .
echo "node: $HOSTNAME" .
echo "jobid: $SLURM_JOB_ID" jobid .
module ~ Linux . GPU , GPU (CPU Partition ) .
Chapter3. module envrionment .
python -m jupyter lab $HOME \ --ip=0.0.0.0 \ --no-browser Jupyter notebook
.

```

## 2.2

, Python .  
, terminal sbatch . (job) .  
(job) ID .

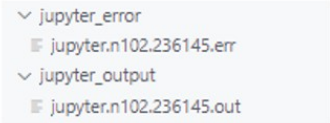
```

sbatch filename.sh # ex) sbatch jupyter.sh

```

(job) , STDOUT OUT .  
OUT , Partition (job) . terminal squeue ,  
ID .  
n001, n002 ... , ( *Resources, Priority* ) .  
, .  
Partition Partition cpus-per-task, gpu Partition (job) .  
ERR , Jupyter Notebook .

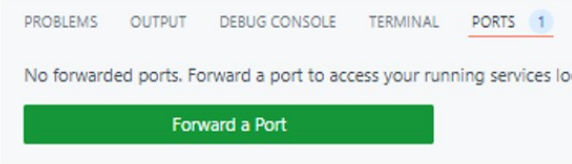
2.3



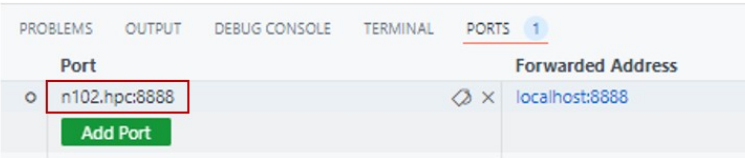
1. STDERR ERR

```
jupyter.n102.236145.err ×
jupyter_error > jupyter.n102.236145.err
1
2 Note: the module "CUDA/11.2.2" cannot be unloaded because it was not loaded.
3
4 [I 2024-10-18 15:20:24.314 ServerApp] jupyter_lsp | extension was successfully linked.
5 [I 2024-10-18 15:20:24.317 ServerApp] jupyter_server_terminals | extension was successfully linked.
6 [I 2024-10-18 15:20:24.321 ServerApp] jupyterlab | extension was successfully linked.
7 [I 2024-10-18 15:20:24.324 ServerApp] notebook | extension was successfully linked.
8 [I 2024-10-18 15:20:24.717 ServerApp] notebook_shim | extension was successfully linked.
9 [I 2024-10-18 15:20:24.745 ServerApp] notebook_shim | extension was successfully loaded.
10 [I 2024-10-18 15:20:24.747 ServerApp] jupyter_lsp | extension was successfully loaded.
11 [I 2024-10-18 15:20:24.748 ServerApp] jupyter_server_terminals | extension was successfully loaded.
12 [I 2024-10-18 15:20:24.751 LabApp] JupyterLab extension loaded from /home1/ssu/miniconda3/envs/ubai/lib/python3.10/site-packages/jupyterlab
13 [I 2024-10-18 15:20:24.751 LabApp] JupyterLab application directory is /gpfs/home1/ssu/miniconda3/envs/ubai/share/jupyter/lab
14 [I 2024-10-18 15:20:24.751 LabApp] Extension Manager is 'pypi'.
15 [I 2024-10-18 15:20:24.774 ServerApp] jupyterlab | extension was successfully loaded.
16 [I 2024-10-18 15:20:24.777 ServerApp] notebook | extension was successfully loaded.
17 [I 2024-10-18 15:20:24.777 ServerApp] Serving notebooks from local directory: /home1/ssu
18 [I 2024-10-18 15:20:24.777 ServerApp] Jupyter Server 2.14.2 is running at:
19 [I 2024-10-18 15:20:24.777 ServerApp] http://n102.hpc:8888/lab?token=d0069066c818d0310ac0a0ce7bd5513e04bdf02a4a4657df
20 [I 2024-10-18 15:20:24.777 ServerApp] http://127.0.0.1:8888/lab?token=d0069066c818d0310ac0a0ce7bd5513e04bdf02a4a4657df
21 [I 2024-10-18 15:20:24.777 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
22 [C 2024-10-18 15:20:24.781 ServerApp]
```

2.



3. VScode PORTS , Forward a Port



- 4. 2 Port , Open in Browser .
- 5. Jupyter .

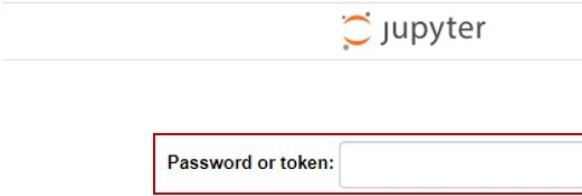
2.4 Jupyter Notebook

Jupyter , . Jupyter Notebook .

1.



STDERR ERR Token .



Token authentication is enabled

- Token . Token , Password .
- 1 , Token .

2.

## Setup a Password

You can also setup a password by entering your token below:

Token

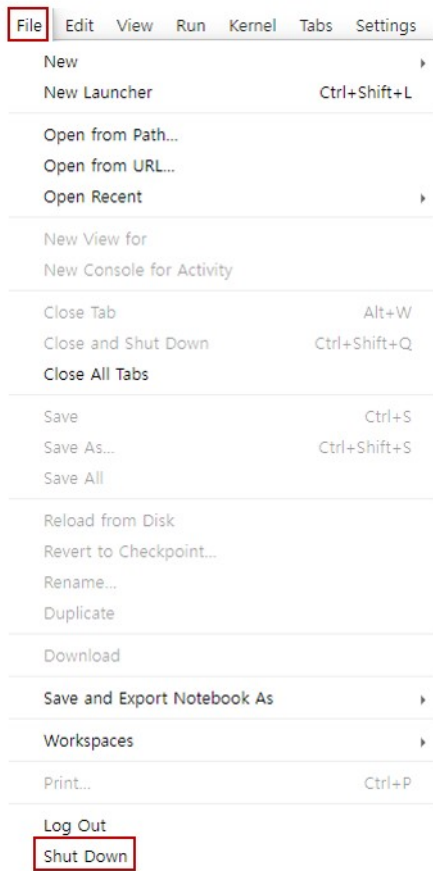
New Password

Log in and set new password

Token , Token New Password Password .  
2 , Password .  
Jupyter Notebook .

## 2.5 Jupyter Notebook

Jupyter Notebook **File** → **Shut Down** , VScode terminal **scancel**  
**{job\_ID}** . , **job\_ID** **job\_ID** .  
(job) .



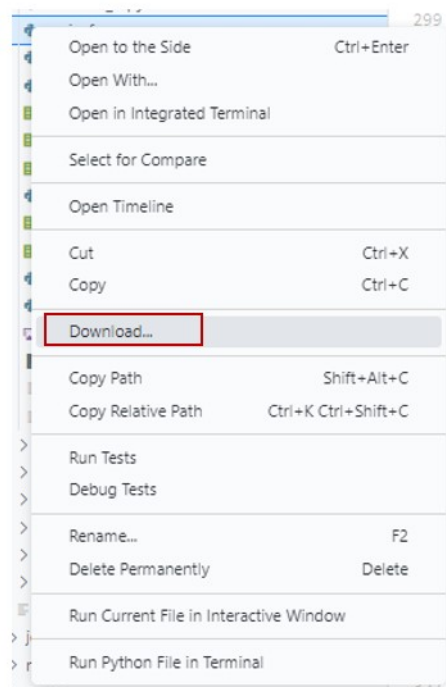
## 7 Chapter5.

UBAI            SSD        100GB        .  
※                    (*ubaisysadmin@uos.ac.kr*)                    1TB        .

1.

VScode        .

7.0.1 1.1



Download        .



## 7.0.2 1.2

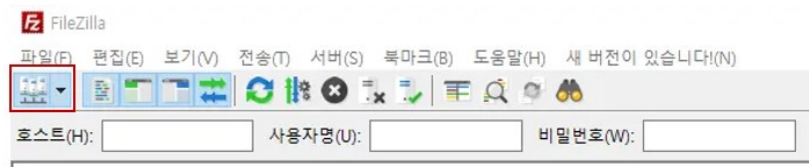
, (Explorer) .

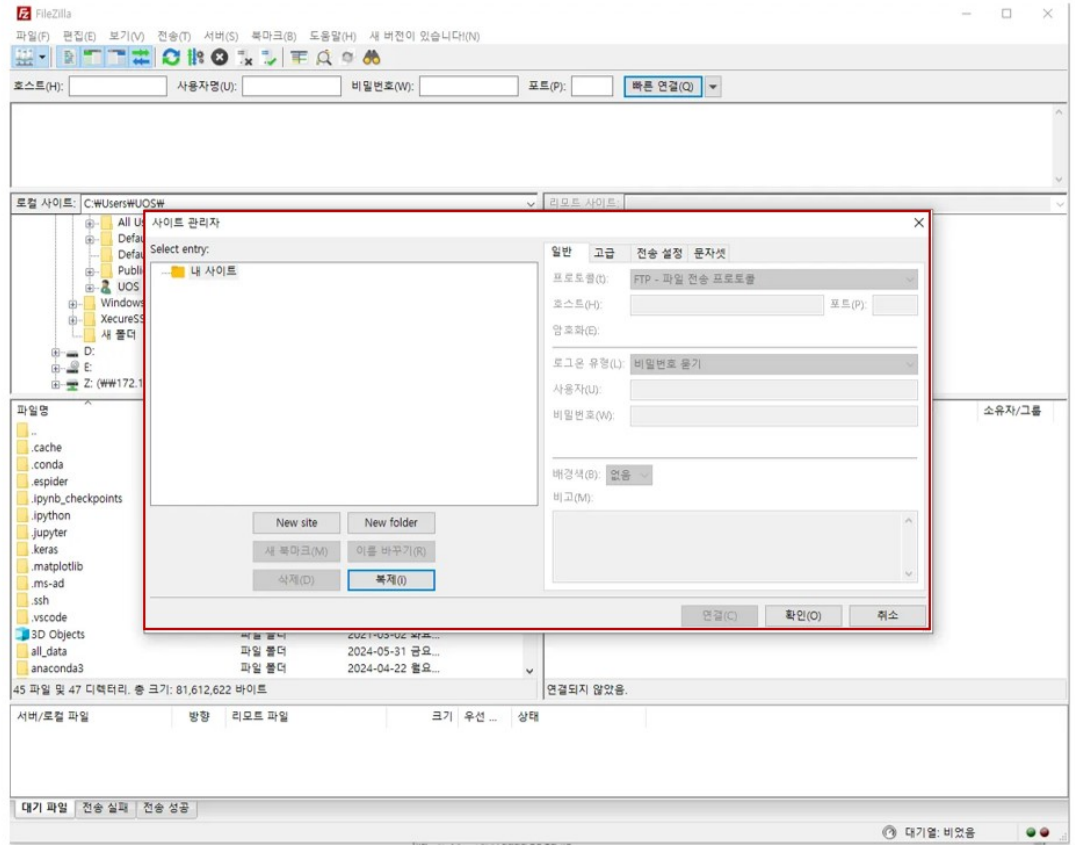
## 2.

. FileZilla . FileZilla .  
FileZilla , .

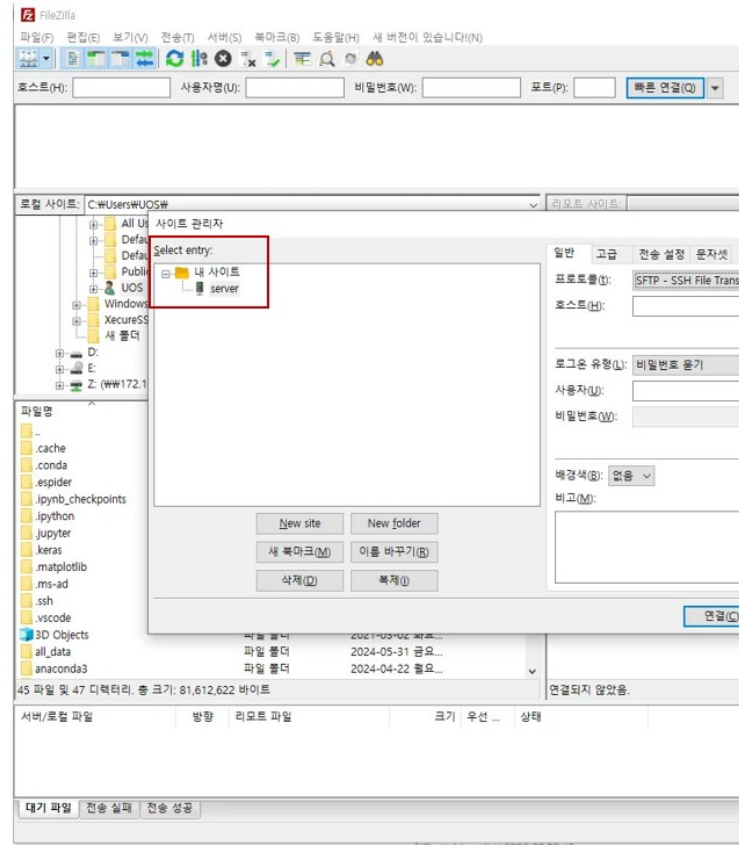
## 7.0.1 2.1

1. .

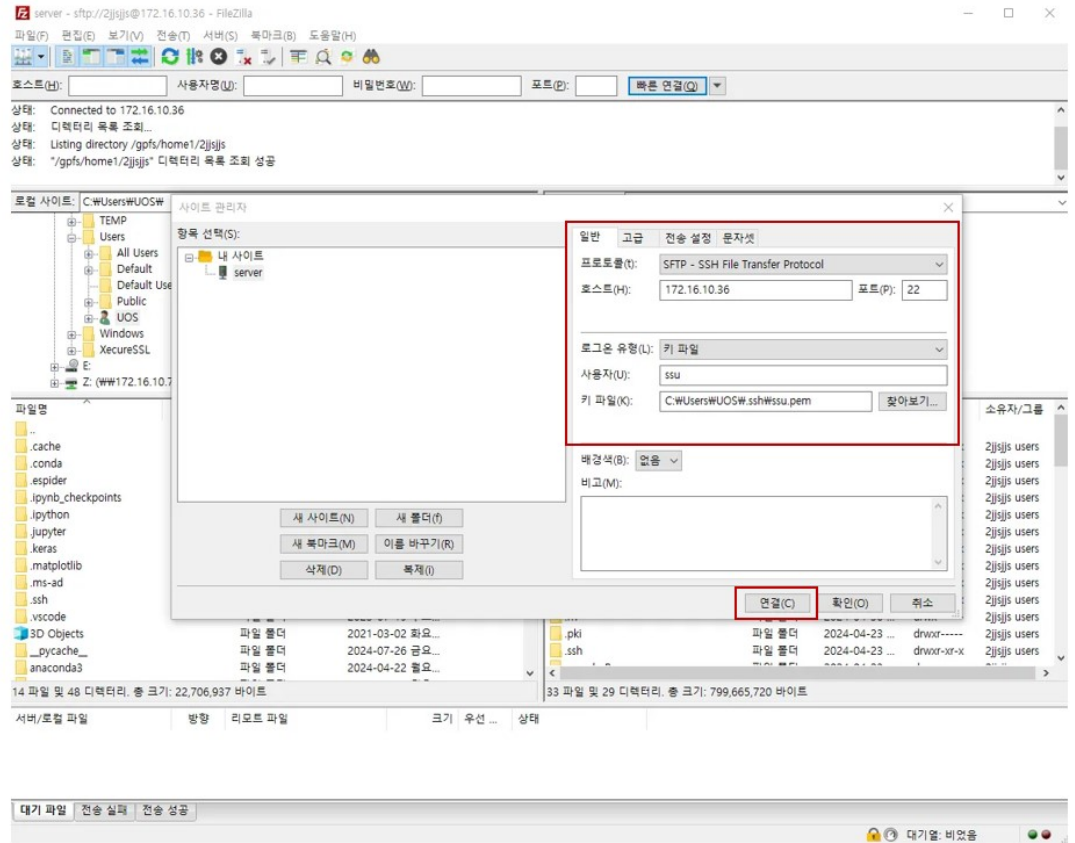




2. New site

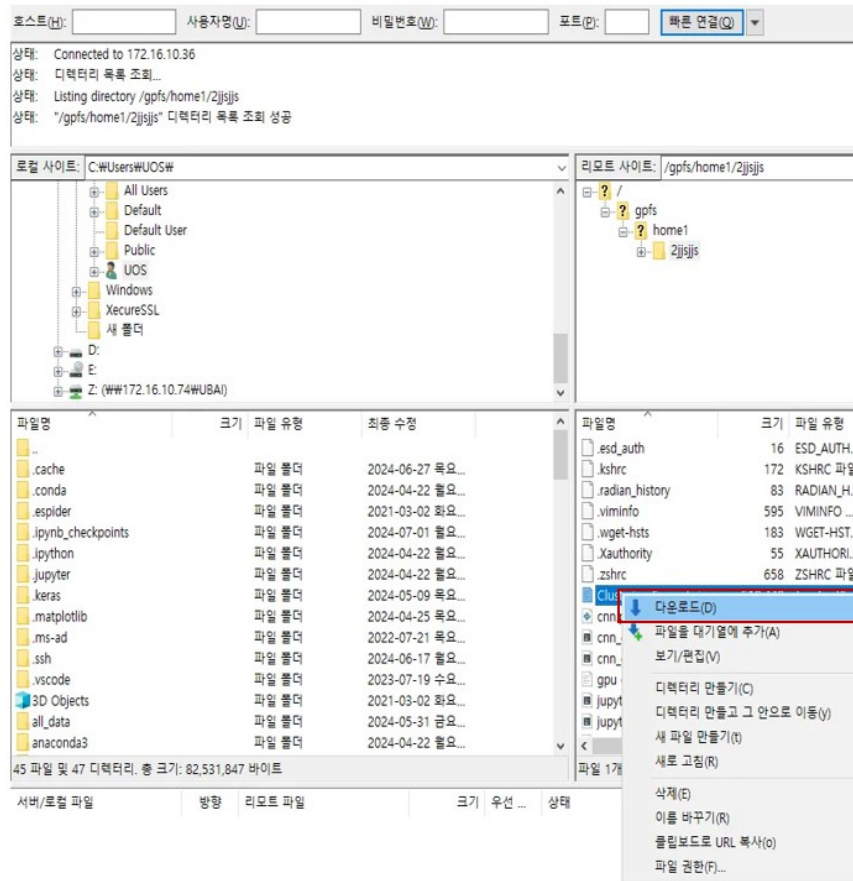


### 3. “SFTP - SSH File Transfer Protocol” .



4.

- : SFTP
- : IPv4
- : 22
- :
- : (ex. ubuntu)
- : (.pem)
-



( )

## 7.0.2 2.2

