

# GSDS Cluster Manual

Graduate School of Data Science in SNU

Manual Ver 1.2

2021. 09. 23.



THUNDER Research Group  
Seoul National University  
서울대학교 천둥 연구실



# Intro

## Basic Rules

GSDS Cluster는 GSDS 전임 교수 및 소속 연구원/학생을 위한 Cluster입니다. Storage에는 연구를 위한 대외비 내용이 저장되어 있을 수 있으므로, **외부인의 사용을 엄격히 금지**합니다. (**GSDS Cluster is NOT allowed for the use of people outside GSDS.**)

**비밀번호 등 보안을 철저히 유지**해주시기 바라며, 특히 외부인 사용 등으로 인하여 본인 계정이 클러스터 외부 침입 및 해킹의 소스가 되지 않도록 각별히 유의해주시기 바랍니다. (**Create strong passwords to protect your data from potential security breach**)



## 매뉴얼 개정 규칙

- 클러스터 현황, 노드별 용도 및 정책과 같은 주요한 변경점이 있는 경우, 버전명의 main number를 변경 (예시: GSDS Cluster Manual\_V1.0, 2.0, ...)
- FAQ, 주의사항과 같은 사소한 변경점이 있는 경우, 버전명의 sub number를 변경 (예시: GSDS Cluster Manual\_V1.1, 1.2, 1.3, ...)
- 매뉴얼 개정 시 GSDS Slack을 통해 개정본이 배포됩니다. 클러스터 담당자는 연구실 내 GSDS Slack 접속이 불가능한 인원에게 매뉴얼을 공유해주시기 바랍니다.



# 목차

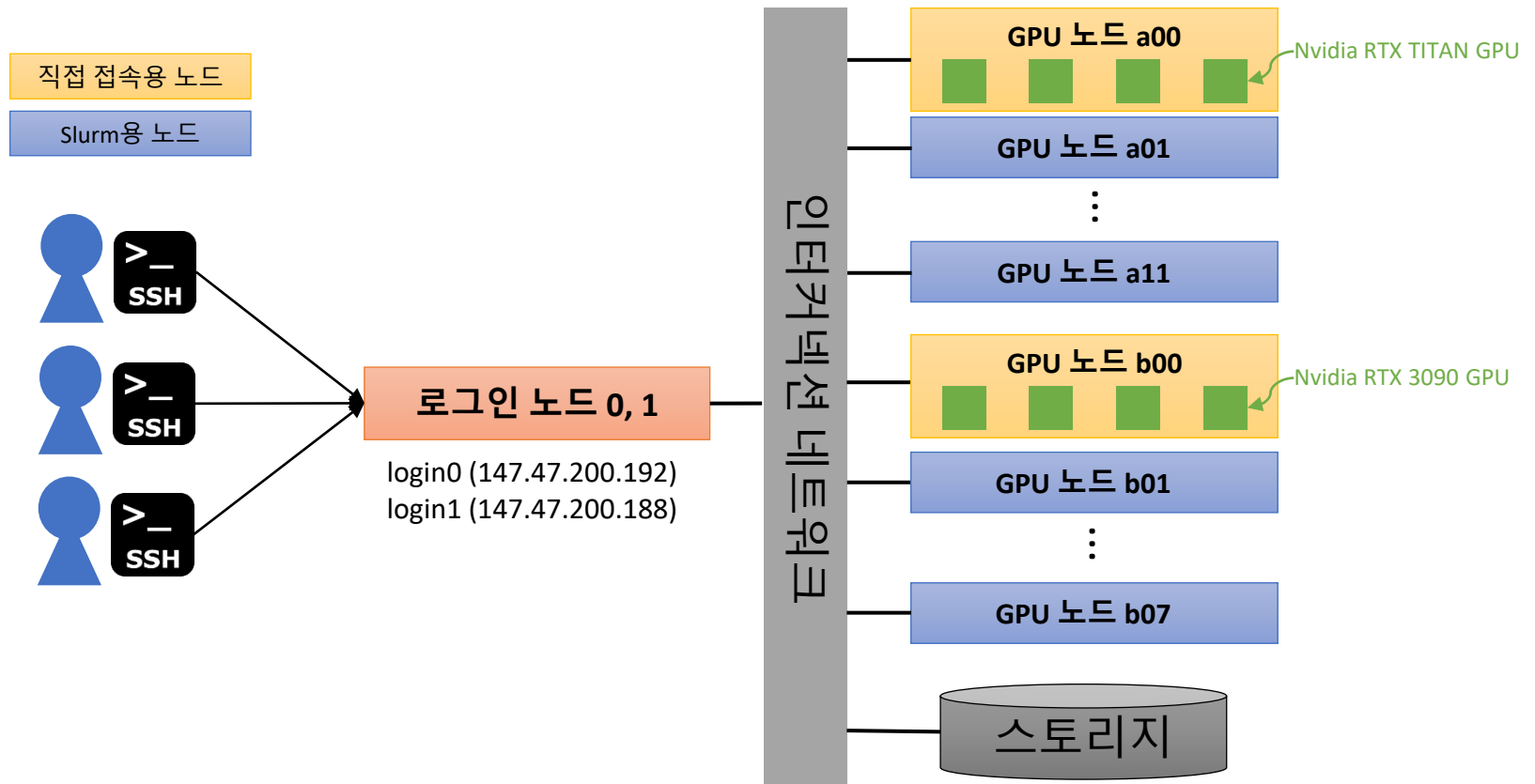
1. 클러스터 현황
2. 노드별 용도 및 정책
3. Linux 사용 방법
4. Slurm 사용 방법
5. Jupyter 사용 방법
6. FAQ
7. 주의사항
8. 각종 신청 양식
9. 매뉴얼 개정 History



# 1. 클러스터 현황

# GSDS 클러스터 현황

- 로그인 노드 2대, GPU 노드 20대 및 스토리지 서버들로 구성된 클러스터

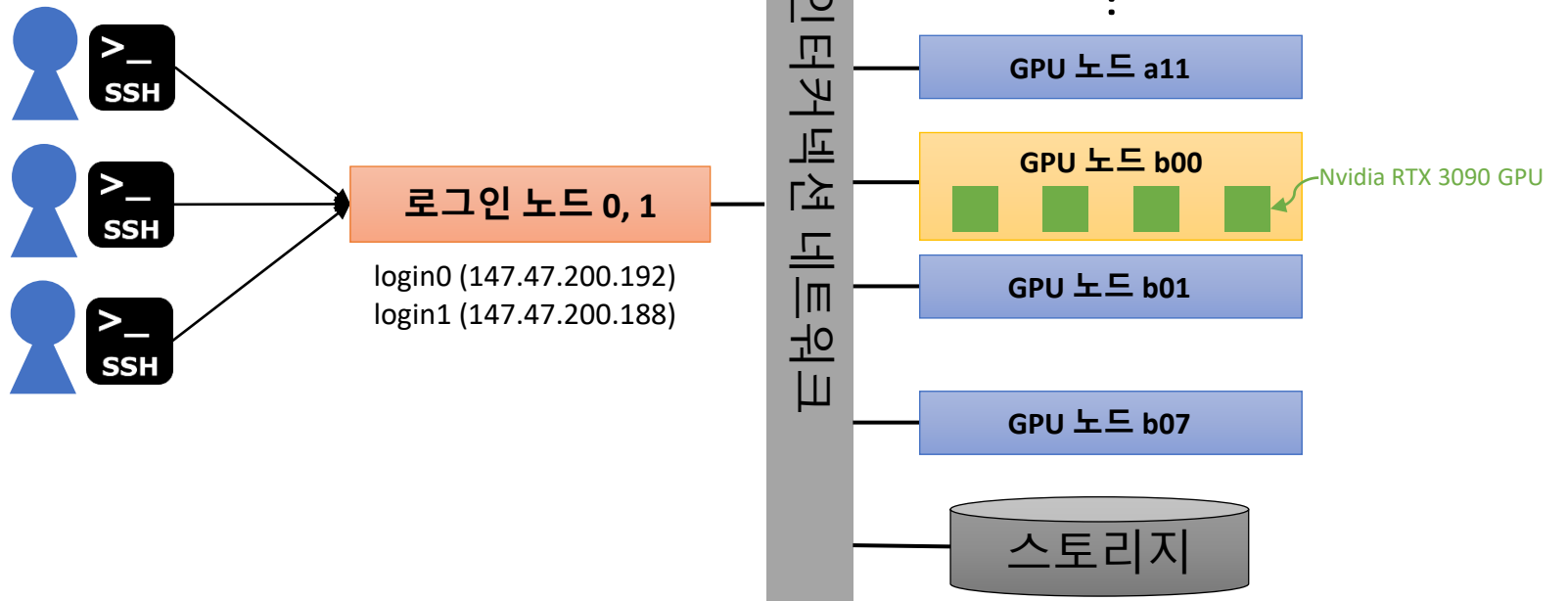


## 2. 노드별 용도 및 정책



# 노드 접속 및 활용 방법

- 총 3가지 활용 방법이 있음
  - 직접 접속하여 로그인 노드 활용
  - 직접 접속하여 GPU 노드 활용
  - Slurm을 통하여 GPU 노드 활용



# GPU 노드별 용도 및 정책

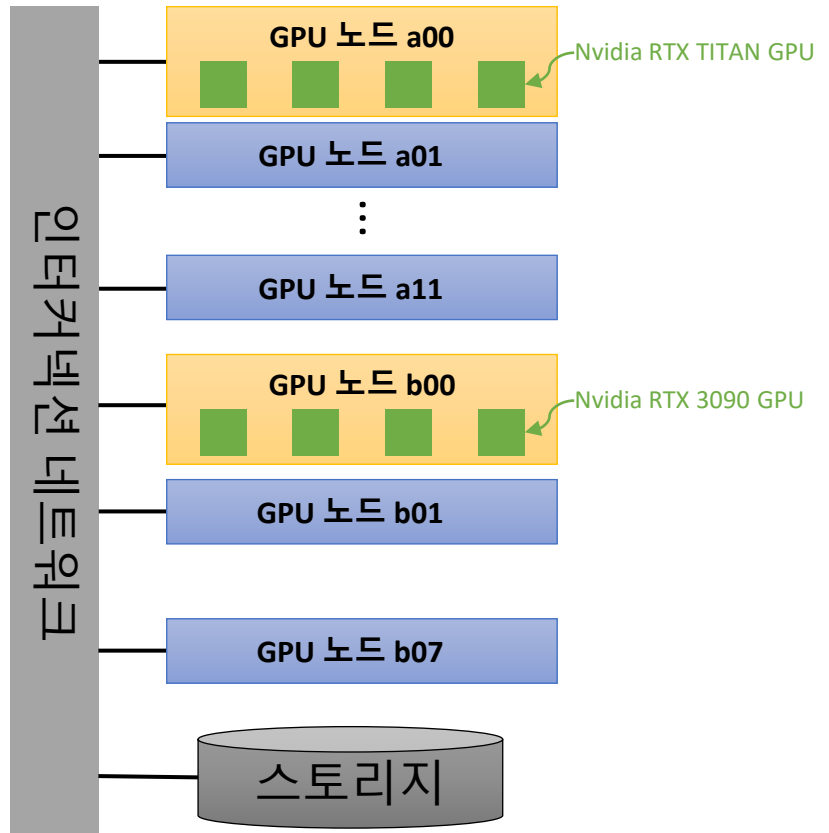
- GPU 노드는 직접 접속 (SSH)이 허용되는 노드 2개와 Slurm만 활용 가능한 노드 18개로 구성됨

## 직접 접속용 노드 (a00, b00)

- 개발 및 디버깅 시에만 활용 가능
- CPU 시간 제한 15분

## Slurm용 노드 (a01-a11, b01-b07)

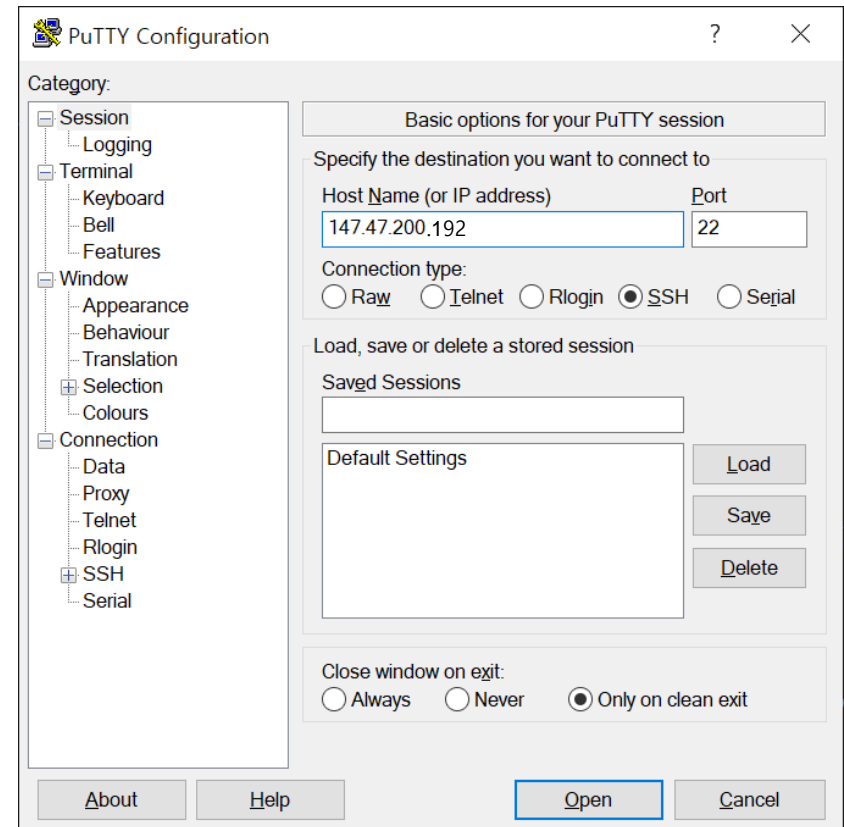
- 딥러닝 training 등 무거운 작업
- CPU 시간 제한 12시간 (default =1시간)
- Slurm으로 노드 할당 받아 SSH처럼 사용 가능



### 3. Linux 사용 방법

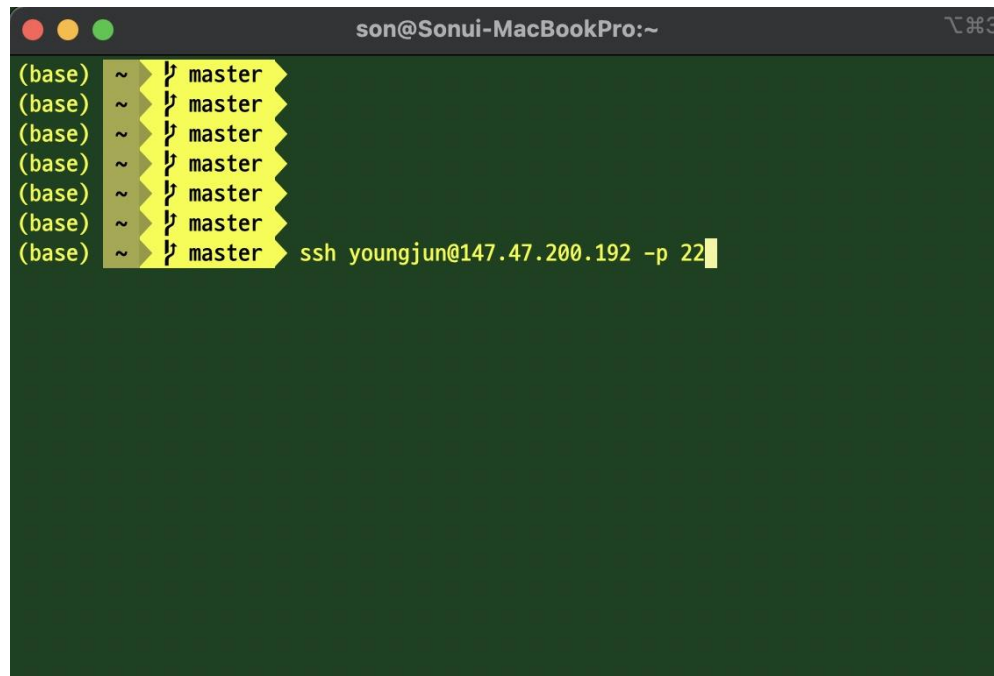
# Accessing a Remote Server (Windows)

- Open a PuTTY client
- IP address:  
147.47.200.192 or 147.47.200.188
- Port: 22 (using SNU network)  
or 22554 (otherwise)
- Connection type: SSH
- Sessions can be saved with  
a specified name



# Accessing a Remote Server (Mac)

- Open a terminal window
- Access a remote server using SSH

A screenshot of a macOS terminal window titled 'son@Sonui-MacBookPro:~'. The window has a dark green background. On the left side, there is a vertical stack of yellow rectangular highlights, each containing the text '(base) ~' followed by a yellow arrow pointing right. The main text in the terminal is 'ssh youngjun@147.47.200.192 -p 22' with a white cursor at the end. The window's title bar shows standard macOS window controls (red, yellow, green buttons) and the title 'son@Sonui-MacBookPro:~'.

```
(base) ~ } master  
(base) ~ } master  
(base) ~ } master  
(base) ~ } master  
(base) ~ } master  
(base) ~ } master  
(base) ~ } master  
(base) ~ } master ssh youngjun@147.47.200.192 -p 22
```



# Accessing a Remote Server (Linux)

- Open a terminal window
- Access a remote server using SSH

```
[C:\~]$ ssh kyusu@147.47.200.192:22554

Connecting to 147.47.200.192:22554...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-72-generic x86_64)

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

System information as of Sun Sep 19 22:48:52 KST 2021

System load:  0.0               Users logged in:      5
Usage of /:   1.7% of 467.96GB  IP address for enp68s0f0: 147.47.200.192
Memory usage: 1%               IP address for enp68s0f1: 192.168.0.10
Swap usage:   0%               IP address for ib0:     192.168.10.10
Processes:   1490

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

279 packages can be updated.
175 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```



# Change the Password

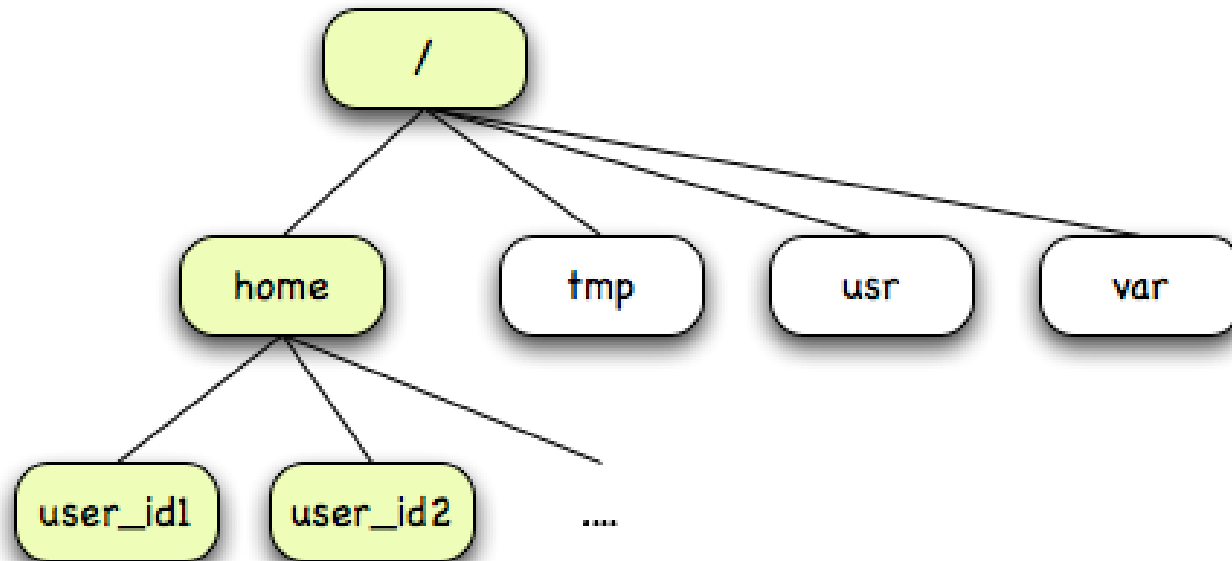
- Logging in & Changing Password
  - Login using your ID
  - **Change your password immediately**

```
Last login: Sun Sep 19 22:41:53 2021 from 123.215.20.251
(base) kyusu@login0:~$ passwd
Enter login(LDAP) password:
New password:
BAD PASSWORD: The password is shorter than 8 characters
New password:
Retype new password:
LDAP password information changed for kyusu
passwd: password updated successfully
```



# The Directory Structure

- All files are grouped together in the directory structure
- The top of the directory hierarchy is traditionally called root





## 4. Slurm 사용 방법

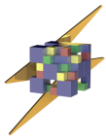
# Slurm이란 무엇인가?

- Slurm은 작업 스케줄러의 한 종류로, 클러스터 시스템 상에서 작업을 관리하기 위한 프로그램임.
- 로그인 노드에서 slurm 스케줄러에 작업을 제출하면, 비어 있는 계산 노드 중 하나를 할당 받아 작업이 실행됨
- 다음의 경우 반드시 Slurm을 사용해야 함
  - 개발용 노드의 제한 시간 (15분)을 넘는 작업을 실행하는 경우
  - Jupyter Notebook을 사용하는 경우



## Slurm 공부에 도움이 되는 주요 명령어

	Link
srun	<a href="https://slurm.schedmd.com/srun.html">https://slurm.schedmd.com/srun.html</a>
sbatch	<a href="https://slurm.schedmd.com/sbatch.html">https://slurm.schedmd.com/sbatch.html</a>
salloc	<a href="https://slurm.schedmd.com/salloc.html">https://slurm.schedmd.com/salloc.html</a>
sinfo	<a href="https://slurm.schedmd.com/sinfo.html">https://slurm.schedmd.com/sinfo.html</a>
squeue	<a href="https://slurm.schedmd.com/squeue.html">https://slurm.schedmd.com/squeue.html</a>
scancel	<a href="https://slurm.schedmd.com/scancel.html">https://slurm.schedmd.com/scancel.html</a>



# 작업 제출

- 작업 제출은 login0, login1 노드에서 가능
- 작업 제출 방법은 두 가지가 있음
  1. sbatch 커맨드로 작업 제출 가능. 작업 실행 커맨드와 설정을 담은 run.sh 스크립트를 준비하고, sbatch run.sh 커맨드로 작업을 제출하면 됨.  
자세한 설명은 [링크](#) 에서 확인 가능.
  2. Slurm을 직접 접속처럼 사용 (launch-shell)



## 4. Slurm 사용 방법

① sbatch

② launch-shell

# 작업 제출 방법 ① - sbatch

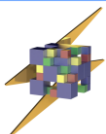
- run.sh의 내용을 아래와 같이 구성
  - single-node+ single-GPU, 아나콘다 가상환경, Python

```
#!/bin/bash
```

```
#SBATCH --job-name=example      # Submit a job named "example"
#SBATCH --nodes=1                # Using 1 node
#SBATCH --gres=gpu:1            # Using 1 GPU
#SBATCH --time=0-00:02:00       # 2 minute timelimit
#SBATCH --mem=16000MB           # Using 16GB memory
#SBATCH --cpus-per-task=8       # Using 8 cpus per task (srun)

source /home/${USER}/.bashrc    # Initiate your shell environment
conda activate <your torch env> # Activate your conda environment

srun python <your python script> <args...>
```



# 작업 제출 방법 ① - sbatch

- run.sh의 내용을 아래와 같이 구성
  - single-node + multi-GPU, 아나콘다 가상환경, Python

```
#!/bin/bash
```

```
#SBATCH --job-name=example      # Submit a job named "example"
#SBATCH --nodes=1                # Using 1 node
#SBATCH --gres=gpu:2             # Using 2 GPU
#SBATCH --time=0-00:02:00        # 2 minute timelimit
#SBATCH --mem=16000MB            # Using 16GB memory
#SBATCH --cpus-per-task=8        # Using 8 cpus per task (srun)

source /home/${USER}/.bashrc     # Initiate your shell environment
conda activate <your torch env>  # Activate your conda environment

srun python <your python script> <args...>
```



## 작업 제출 방법 ① - sbatch

- run.sh의 내용을 아래와 같이 구성
  - single-node + single-GPU, 바이너리

```
#!/bin/bash
```

```
#SBATCH --job-name=example      # Submit a job named "example"
#SBATCH --nodes=1                # Using 1 node
#SBATCH --gres=gpu:1             # Using 1 GPU
#SBATCH --time=0-00:02:00        # 2 minute timelimit
#SBATCH --mem=16000MB            # Using 16GB memory
#SBATCH --cpus-per-task=8        # Using 8 cpus per task (srun)

source /home/${USER}/.bashrc     # Initiate your shell environment

srun <your program>
```





# 작업 제출 방법 ① - sbatch

- run.sh의 내용을 아래와 같이 구성
  - single-node + multi-GPU, 바이너리

```
#!/bin/bash
```

```
#SBATCH --job-name=example      # Submit a job named "example"
#SBATCH --nodes=1                # Using 1 node
#SBATCH --gres=gpu:3             # Using 3 GPU
#SBATCH --time=0-00:02:00       # 2 minute timelimit
#SBATCH --mem=16000MB           # Using 16GB memory
#SBATCH --cpus-per-task=8        # Using 8 cpus per task (srun)

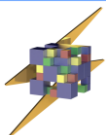
source /home/${USER}/.bashrc    # Initiate your shell environment

srun <your program>
```



## 작업 제출 방법 ① - sbatch

- 따로 설정을 하지 않은 경우, 작업의 출력(stdout 및 stderr)은 모두 *slurm-%j.out* 파일에 저장됨. (%j는 job id). #SBATCH --output 옵션으로 출력이 저장될 파일을 지정할 수 있음.
- 모든 작업의 기본 timelimit 은 1시간으로 설정되어 있음. 이는 실수로 제출된 작업이 자원을 너무 오래 점유하지 않도록 하기 위함임. #SBATCH --time 옵션으로 이 제한을 넘겨 설정할 수 있음 (최대 12시간).



## 4. Slurm 사용 방법

① sbatch

② launch-shell

## 작업 제출 방법 ② - launch-shell

- launch-shell은 GPU 노드에 직접 접속하는 것처럼 해주며, 내부적으로는 slurm 스케줄러를 통해 노드를 할당 받은 것임
  - 아래 예시 화면을 보면 *kyusu@login0* → *kyusu@b00* 로 변경된 것을 확인할 수 있음
- launch-shell [#GPUs] [timelimit in minutes (optional)] [slurm partition (optional)]
  - [#GPUs]: 0 – 4 사이의 숫자를 입력 (0 입력 시 CPU 할당)
  - [timelimit]: 1 – 720 사이의 숫자를 입력 (그 이상 숫자 입력해도 12시간 후 강제 kill)
  - Slurm partition: titan 또는 3090 입력
- 노드 접속 후 (아래의 경우 b00), 필요에 따라 가상환경 activate하면 됨

```
(upca) kyusu@login0:~$ launch-shell 1 120 3090
# of GPUS: 1
timelimit in minutes: 120
Slurm Partition: 3090
(base) kyusu@b00:~$
```



## 작업 현황 확인

- `squeue` 커맨드로 실행 또는 대기중인 작업들의 현황을 확인
- `-u` 옵션으로 자신의 작업 현황만 추려내서 확인 가능

```
kyusu@login0:~$ squeue -u kyusu
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	ODELIST(Reason)
292	3090	jupyter	kyusu	R	15:51	1	b00

- 리스트의 ST 열이 작업의 상태를 나타냄
  - R (Running) - 작업이 자원을 할당 받고 실행 중.
  - PD (Pending) - 작업이 자원을 할당 받기를 기다리는 중.
  - CG (Completing) - 작업이 완료되고 끝나는 중.



## 제출한 작업 취소

- 먼저 `queue` 커맨드를 통해 취소할 작업의 job id를 알아내고,  
`scancel <job id>` 커맨드로 작업을 취소

```
kyusu@login0:~$ queue -u kyusu
      JOBID PARTITION    NAME    USER ST      TIME  NODES NODELIST(REASON)
      292      3090  jupyter  kyusu  R    15:51      1 b00
kyusu@login0:~$ scancel 292
kyusu@login0:~$ queue
      JOBID PARTITION    NAME    USER ST      TIME  NODES NODELIST(REASON)
kyusu@login0:~$ █
```

- u 옵션으로 자신의 모든 작업을 한번에 취소 가능

```
(base) kyusu@login0:~$ queue -u kyusu
      JOBID PARTITION    NAME    USER ST      TIME  NODES NODELIST(REASON)
      2055      3090    shell  kyusu  R     0:12      1 b01
      2056      3090  jupyter  kyusu  R     0:11      1 b01
      2053      titan  jupyter  kyusu  R     1:25      1 a01
(base) kyusu@login0:~$ scancel -u kyusu
(base) kyusu@login0:~$ queue -u kyusu
      JOBID PARTITION    NAME    USER ST      TIME  NODES NODELIST(REASON)
```



# 자원 현황 확인

- sinfo (자세한 설명은 [링크](#) 참조)
  - sinfo 커맨드로 노드 및 GPU 사용 현황 확인

```
kyusu@login0:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
titan      up       10:00:00    12   idle a[00-11]
3090*      up       10:00:00     8   idle b[00-07]
```

- 다음과 같이 좀 더 자세한 정보 출력도 가능함

```
kyusu@login0:~$ sinfo -N -o "%.20N %.15C %.10t %.10m %.15P %.30G"
      NODELIST  CPUS(A/I/O/T)  STATE  MEMORY  PARTITION  GRES
      a00        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a01        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a02        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a03        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a04        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a05        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a06        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a07        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a08        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a09        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a10        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      a11        0/64/0/64    idle   400000   titan     gpu:titan:4(S:0-1)
      b00        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b01        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b02        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b03        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b04        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b05        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b06        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
      b07        0/128/0/128   idle   400000   3090*     gpu:rtx_3090:4(S:0-1)
```



# 5. Jupyter Notebook 사용 방법



## 5. Jupyter Notebook 사용 방법

- Jupyter Notebook을 사용하기 위하여 다음의 과정 필요
  1. launch-jupyter 커맨드 실행
    - 아나콘다 및 주피터 설치 (최초 실행 시 자동으로 설치됨)
    - Slurm 작업 제출 (자동으로 제출됨)
  2. 작업 배정여부 확인 (squeue) – 작업 제출한 shell이 아닌 곳에서 해야 함
  3. 출력되는 링크를 인터넷 창에 붙여넣기
- Jupyter Notebook 사용 후 반드시 작업을 **3가지 방법 중 하나로 종료**해주세요.
  - **Jupyter 화면 우측 상단의 Quit 클릭 (추천하는 방식)**
  - launch-jupyter 실행 화면에서 Ctrl + C 두 번 누르기
  - scancel {JOBID} 커맨드로 종료
- 변경된 Jupyter Notebook 접속 방법은 기존 대비 아래와 같은 장점이 있음
  - Linux와 storage를 공유함
  - 본인이 생성한 가상 환경 사용 가능
  - GPU resource utilization 개선



# 5. Jupyter Notebook 사용 방법

- ① launch-jupyter
- ② 작업 배정여부 확인
- ③ 출력되는 링크 붙여넣기

# Jupyter Notebook 실행: launch-jupyter

- 최초 실행 시: launch-jupyter 0
- 최초 실행 시 Anaconda 및 Jupyter를 자동으로 설치해주며, 설치하는 접속 환경에 따라 최대 30분 소요

```
kyusu@login0:~$ launch-jupyter 0
Cheking anaconda is installed... Not installed.
Installing anaconda... Please wait.
PREFIX=/home/kyusu/anaconda
Unpacking payload ...
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/kyusu/anaconda

added / updated specs:
- _ipyw_jlab_nb_ext_conf==0.1.0=py38_0
- _libgcc_mutex==0.1=main
- alabaster==0.7.12=pyhd3eb1b0_0
- anaconda-client==1.7.2=py38_0
- anaconda-navigator==2.0.3=py38_0
```



# Jupyter Notebook 실행: launch-jupyter (Cont'd)

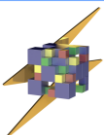
- launch-jupyter [#GPUs] [conda environment (optional)] [timelimit in minutes (optional)] [slurm partition (optional)]
  - [#GPUs]: 0 – 4 사이의 숫자를 입력 (0 입력 시 CPU 할당)
  - [conda environment]: 입력한 가상환경이 자동으로 activate 되어 Jupyter Notebook이 실행됨
  - [timelimit]: 1 – 720 사이의 숫자를 입력 (그 이상 숫자 입력해도 12시간 후 강제 kill)
  - Slurm partition: titan 또는 3090 입력

```
(base) kyusu@login0:~$ launch-jupyter 1 upca 120 3090
Checking anaconda is installed... OK.
Checking jupyter is installed... OK.
# of GPUs: 1
timelimit in minutes: 120
conda env: upca
Slurm Partition: 3090

!!
!! Replace 127.0.0.1 with 147.47.200.192 in the url below
!!

[W 2021-09-20 16:20:22.976 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[W 2021-09-20 16:20:23.619 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-20 16:20:23.620 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-20 16:20:23.620 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-20 16:20:23.620 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2021-09-20 16:20:23.630 LabApp] JupyterLab extension loaded from /home/kyusu/anaconda/envs/upca/lib/python3.9/site-packages/jupyterlab
[I 2021-09-20 16:20:23.630 LabApp] JupyterLab application directory is /home/kyusu/anaconda/envs/upca/share/jupyter/lab
[I 16:20:23.636 NotebookApp] Serving notebooks from local directory: /home/kyusu
[I 16:20:23.636 NotebookApp] Jupyter Notebook 6.4.4 is running at:
[I 16:20:23.636 NotebookApp] http://b02:31495/?token=b24ce3bd47d96169a5757cf82bded80e077c00fefa03fd31
[I 16:20:23.636 NotebookApp] or http://127.0.0.1:31495/?token=b24ce3bd47d96169a5757cf82bded80e077c00fefa03fd31
[I 16:20:23.636 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:20:23.647 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/kyusu/.local/share/jupyter/runtime/nbserver-83531-open.html
Or copy and paste one of these URLs:
http://b02:31495/?token=b24ce3bd47d96169a5757cf82bded80e077c00fefa03fd31
or http://127.0.0.1:31495/?token=b24ce3bd47d96169a5757cf82bded80e077c00fefa03fd31
```



# Jupyter Notebook 실행: launch-jupyter (Cont'd)

- launch-jupyter 실행 시 유의할 점
  - launch-jupyter [#GPUs] 까지만 입력하는 것은 무방하지만, 중간에 optional argument를 생략하고 *뒤/쪽의 argument*만 작성하면 안 됨
    - 예: launch-jupyter [#GPUs] *[slurm partition (optional)]* (잘못된 실행 예)
  - Optional argument 중 [conda environment]를 입력하여 launch-jupyter를 실행하는 경우, 해당 [conda environment]에 jupyter가 설치되어 있어야 함
    - 설치 방법 1:



# 5. Jupyter Notebook 사용 방법

- ① launch-jupyter
- ② 작업 배정여부 확인
- ③ 출력되는 링크 붙여넣기

## Jupyter Notebook 실행: launch-jupyter

- `squeue` 커맨드로 본인의 작업이 배정되었는지 확인
- `launch-jupyter`를 실행한 shell은 `srunit`이 잡고 있으므로, 작업 배정여부 확인은 다른 shell에서 해야 함

```
kyusu@login0:~$ squeue
      JOBID PARTITION    NAME    USER  ST       TIME  NODES NODELIST(REASON)
       292       3090  jupyter  kyusu   R        4:48      1 b00
```



# 5. Jupyter Notebook 사용 방법

- ① launch-jupyter
- ② 작업 배정여부 확인
- ③ 출력되는 링크 붙여넣기



# Jupyter Notebook 실행: launch-jupyter

- launch-jupyter 실행 후 출력되는 링크를 인터넷 창에 붙여넣기

```
To access the notebook, open this file in a browser:  
file:///home/kyusu/.local/share/jupyter/runtime/nbserver-88535-open.html  
Or copy and paste one of these URLs:  
http://b00:31281/?token=4d575e1f9250814d50507d9f792207954d4f88b4a369e1f9  
or http://127.0.0.1:31281/?token=4d575e1f9250814d50507d9f792207954d4f88b4a369e1f9
```

- 이 때, 두 주소 중 **b00** 혹은 **127.0.0.1** 부분을 **147.47.200.192**로 수정
- 정상 접속되었다면 아래와 같은 화면이 출력됨



The screenshot shows the Jupyter Notebook web interface. At the top left is the Jupyter logo. On the top right are 'Quit' and 'Logout' buttons. Below the logo are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' To the right of this message are 'Upload', 'New', and a refresh icon. The main area is a file browser showing the current directory structure. It includes a breadcrumb path '0 /' and a table of files and folders.

	Name ↓	Last Modified	File size
<input type="checkbox"/>	anaconda	3시간 전	
<input type="checkbox"/>	Anaconda3-2021.05-Linux-x86_64.sh	4달 전	571 MB



# Jupyter Notebook 종료

- Logout을 하는 경우, resource 반납이 이루어지지 않음

이 버튼을 눌러  
작업을 종료!!

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↺

<input type="checkbox"/> 0	Name	Last Modified	File size
<input type="checkbox"/>	anaconda	3시간 전	
<input type="checkbox"/>	Anaconda3-2021.05-Linux-x86_64.sh	4달 전	571 MB

- 사용 완료 후 반드시 아래 3가지 방법 중 하나로 작업 종료 필수!
  - Jupyter 화면 우측 상단의 Quit 클릭 (추천하는 방식)
  - launch-jupyter 실행 화면에서 Ctrl + C 두 번 누르기
  - scancel {JOBID} 커맨드로 종료

```
(base) kyusu@login0:~$ squeue
          JOBID PARTITION    NAME    USER  ST       TIME  NODES NODELIST(REASON)
          296      3090  jupyter  kyusu   R        5:59      1  b00
(base) kyusu@login0:~$ scancel 296
(base) kyusu@login0:~$ squeue
          JOBID PARTITION    NAME    USER  ST       TIME  NODES NODELIST(REASON)
(base) kyusu@login0:~$ █
```



## 6. FAQ

## 6. FAQ

- 스토리지 용량을 늘려주세요.
  - 신청 양식을 참고로 내용을 작성하여 [gsds-hpc1@aces.snu.ac.kr](mailto:gsds-hpc1@aces.snu.ac.kr)로 메일 주세요.
- 비밀번호를 초기화 해주세요.
  - 신청 양식을 참고로 내용을 작성하여 [gsds-hpc1@aces.snu.ac.kr](mailto:gsds-hpc1@aces.snu.ac.kr)로 메일 주세요.
- 계정을 신청하고 싶어요.
  - 신청 링크 (Link)에서 계정을 신청해주세요.

※ 스토리지 증설, 비밀번호 초기화 신청 시 양식에 맞지 않는 경우 대응하지 않습니다.

※ 클러스터 정식 오픈 이후에는, **신청된 계정의 생성을 매주 금요일에만 진행**합니다.



## 6. FAQ

- SSH가 안돼요.
  - SSH는 2개의 GPU 노드 (a00, b00)에만 허용됩니다. 다른 계산 노드에 SSH를 시도하지 않았는지 확인해주세요.
- Docker는 사용할 수 없나요?
  - Docker가 Lustre 파일시스템과 호환이 되지 않습니다. 억지로 돌려도 (ex. 로컬 스토리지에 잡고 돌림) 문제가 너무 많이 발생해서 지원하지 않기로 결정하였습니다.
- GPU 서버 시간제한을 늘려 주세요.
  - Slurm 시간 제한은 제한된 자원을 많은 구성원이 공평하게 사용하기 위한 정책입니다.
  - 시간제한보다 오래 돌려야 하는 모델 트레이닝 등의 작업은 중간중간 checkpoint 를 만들어 저장하고 다시 시작하는 방식으로 실행시켜 주세요



## 7. 주의사항

## 7. 주의사항

- 아래와 같이 클러스터 설정과는 전혀 관련 없는 것들은 클러스터 관리자에게 질문하지 말아 주세요.
  - 아나콘다 설치 어떻게 해요?
  - Python module import 시 에러가 나요.
- 위와 같은 질문은 연구실 별 서버 담당자에게 해주세요.**

교수님	담당자
Wen-Syan Li	이우철
김용대	정지원
김형신	김우중
박현우	박주연
신호필	이윤섭
오민환	박승준
이상학	조동규
이승근	안지용
이재진	손영준
이준석	김주은
차상균	채규욱

(2021년 담당자)



## 7. 주의사항 (Cont'd)

- 보안에 유의해주세요.
  - 패스워드는 별첨의 *패스워드선택및이용가이드.pdf* 를 참고해주세요.
  - 패스워드를 타인 (특히 학내 구성원 외)과 공유하거나 유추하기 쉬운 것으로 설정하지 말아 주세요.

※ *패스워드선택및이용가이드.pdf*에 부합하지 않는 패스워드 선택 혹은 패스워드 타인 공유로 인하여 본인의 계정이 클러스터 외부 침입의 소스가 되는 경우, 향후 클러스터 이용에 불이익이 있을 수 있습니다.





## 7. 주의사항 (Cont'd)

- Slurm에 동시에 너무 많은 작업을 제출하지 말아 주세요.
- 수업용 파티션에 작업을 제출하지 말아 주세요.
- 여러 주피터 노트북을 동시에 생성하면 에러가 발생하니, 한 번에 하나의 주피터 노트북을 생성해주세요.
- launch-jupyter 및 launch-shell은 16코어/100G메모리 로 할당되어 있으므로, 더 많은 resource를 사용하고 싶은 경우 srun, salloc 등을 개인적으로 공부해서 사용하시면 됩니다.



## 7. 주의사항 (Cont'd)

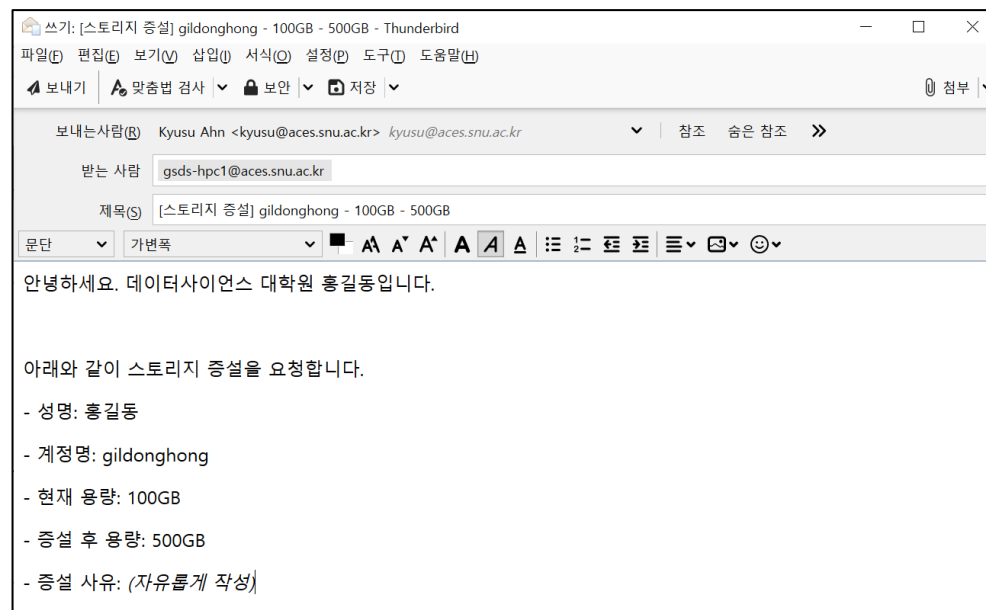
- 본인의 코드가 GPU를 사용하도록 구현되어 있지 않다면 주피터 및 shell 사용 시 (launch-jupyter, launch-shell 실행 시), [#GPUs] argument를 0으로 설정해주세요. CPU만 사용하는데 GPU를 할당 받음으로 인하여, 긴 시간동안 GPU가 필요한 다른 학우들이 GPU를 사용하지 못하는 일이 없도록 서로 배려해주세요.
  - 직접 접속의 경우 nvidia-smi를 통해 GPU usage를 확인할 수 있는데, slurm을 사용하는 GSDS cluster에서 nvidia-smi 사용 불가. GPU usage를 확인할 수 있는 스크립트를 작성 중이며, 추후 배포 예정



## 8. 각종 신청 양식

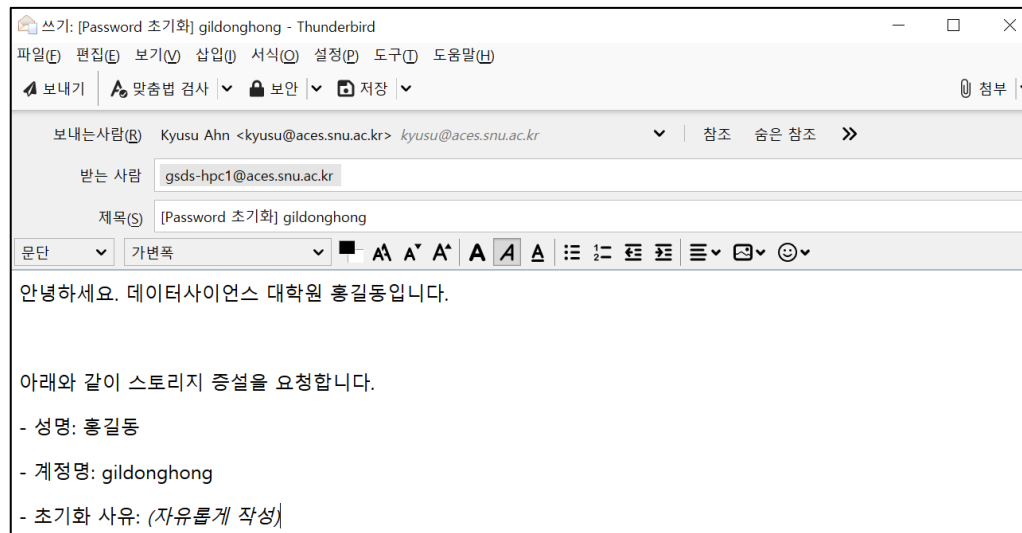
# 스토리지 증설 요청 양식

- 스토리지가 부족한 경우, 아래의 항목을 작성하여 Cluster 관리자에게 메일 발송 시, 자원 현황을 고려하여 스토리지 증설이 가능함
  - 메일 제목: [스토리지 증설] {계정명} - {현재 용량} - {증설 후 용량}
  - 메일 내용: 성명/계정명/현재 용량/증설 후 용량/증설 사유 가 **모두** 포함되어야 함



# Password 초기화 요청 양식

- Password 초기화를 원하는 경우, 아래의 항목을 작성하여 Cluster 관리자에게 메일 발송
  - 메일 제목: [비밀번호 초기화] {계정명}
  - 메일 내용: 성명/계정명/초기화 사유 가 **모두** 포함되어야 함



쓰기: [Password 초기화] gildonghong - Thunderbird

파일(F) 편집(E) 보기(V) 삽입(I) 서식(O) 설정(P) 도구(T) 도움말(H)

보내기 | 맞춤법 검사 | 보안 | 저장 | 첨부

보내는사람(R) Kyusu Ahn <kyusu@aces.snu.ac.kr> kyusu@aces.snu.ac.kr | 참조 숨은 참조 >>

받는 사람 gsds-hpc1@aces.snu.ac.kr

제목(S) [Password 초기화] gildonghong

문단 | 가변폭 | [Rich Text Editor Icons]

안녕하세요. 데이터사이언스 대학원 홍길동입니다.

아래와 같이 스토리지 증설을 요청합니다.

- 성명: 홍길동
- 계정명: gildonghong
- 초기화 사유: (자유롭게 작성)



## 9. 매뉴얼 개정 History

## 8. 매뉴얼 개정 History

- V0.0 노트 포맷 및 개편 후 최초 작성 (2021. 09. 19.)
- V1.0 배포 (2021. 09. 20.)
- V1.1 배포: Jupyter Notebook 접속방법 수정 (2021. 09. 23.)
- V1.2 배포: launch-jupyter 실행 시 conda env 환경 관련 내용 / 주의사항 추가 (2021. 09. 29.)

