

# GSDS C Cluster Manual

Graduate School of Data Science in SNU

Manual Ver 0.1

2022. 08. 10.



THUNDER Research Group  
Seoul National University  
서울대학교 천동 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 금지합니다.** (**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

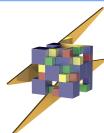
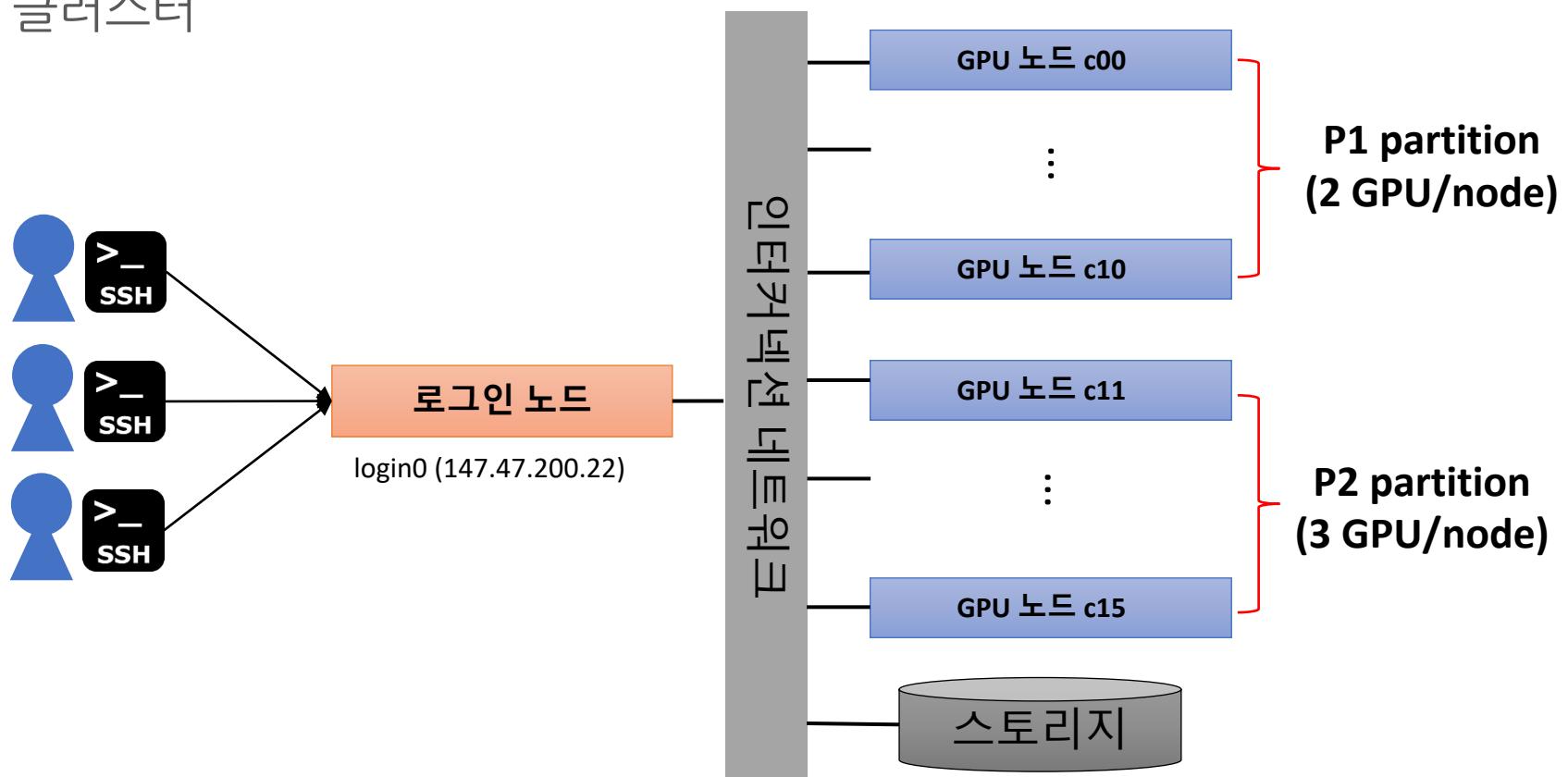
1. 클러스터 현황
2. 노드별 용도 및 정책
3. Linux 사용 방법
4. Slurm 사용 방법
5. Jupyter 사용 방법
6. FAQ
7. 주의사항
8. 각종 신청 양식
9. 매뉴얼 개정 History



# 1. 클러스터 현황

# GSDS 클러스터 현황

- 로그인 노드 1대, GPU 노드 16대 및 스토리지 서버들로 구성된 클러스터



## 2. 노드별 용도 및 정책

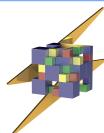
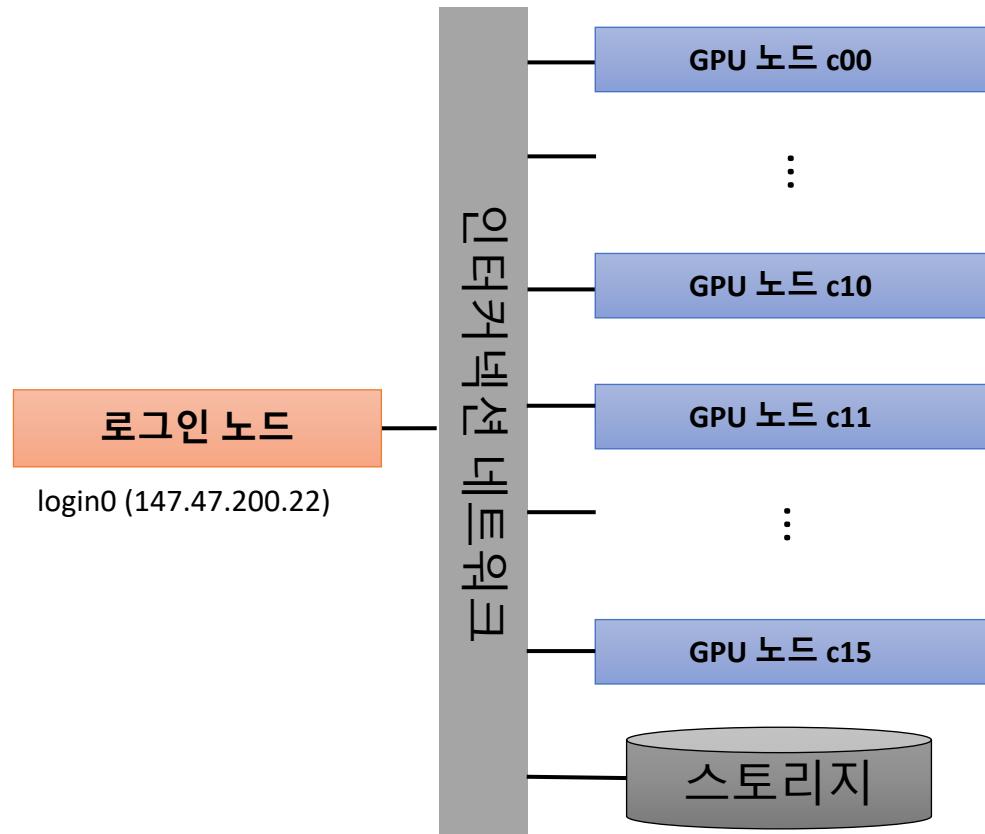
# GPU 노드별 용도 및 정책

## 로그인노드 (login0)

- GPU 노드로 이동하기 위한 용도로만 사용
  - 간단한 file I/O 정도는 가능
- 계산 코드 실행 절대 금지
  - 관리자가 임의로 프로세스 kill 예정

## Slurm용 노드 (c00-c15)

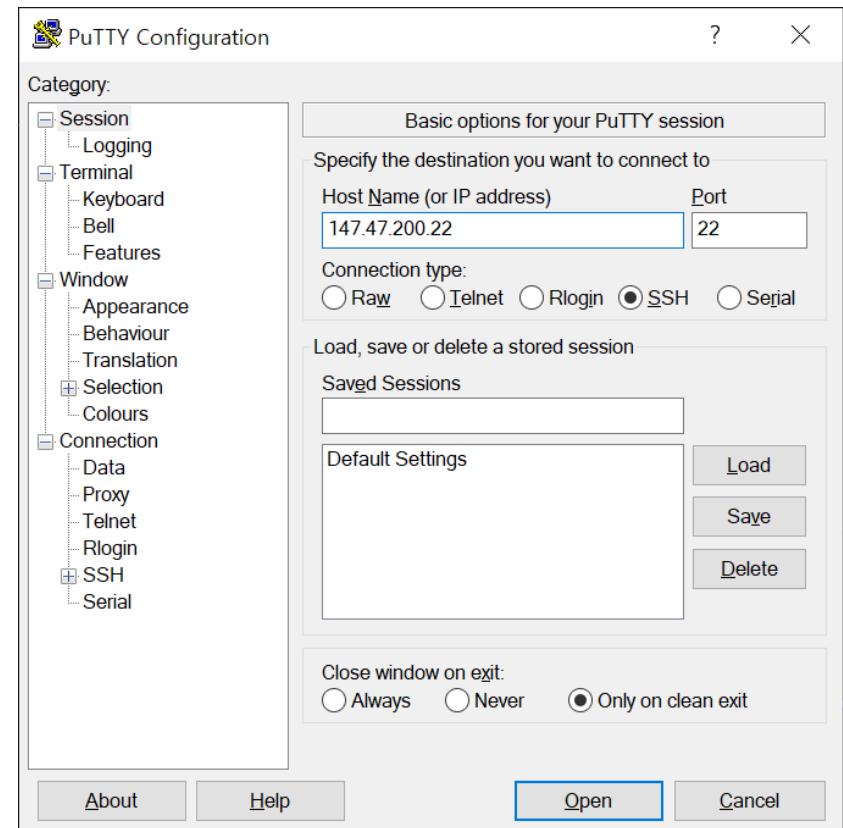
- 딥러닝 training 등 무거운 작업
- CPU 시간 제한 12시간 (default = 1시간)
- Slurm으로 노드 할당 받아 SSH처럼 사용 가능



# 3. Linux 사용 방법

# Accessing a Remote Server (Windows)

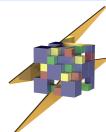
- Open a PuTTy client
- IP address:  
147.47.200.22
- Port: 22 (using SNU network)  
or 22554 (otherwise)
- Connection type: SSH
- Sessions can be saved with  
a specified name



# Accessing a Remote Server (Mac, Linux)

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

```
(base) ~ ➜ master ➜ ssh son@147.47.200.22 -p 22  
son@147.47.200.22's password: ↵
```



# Change the Password

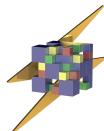
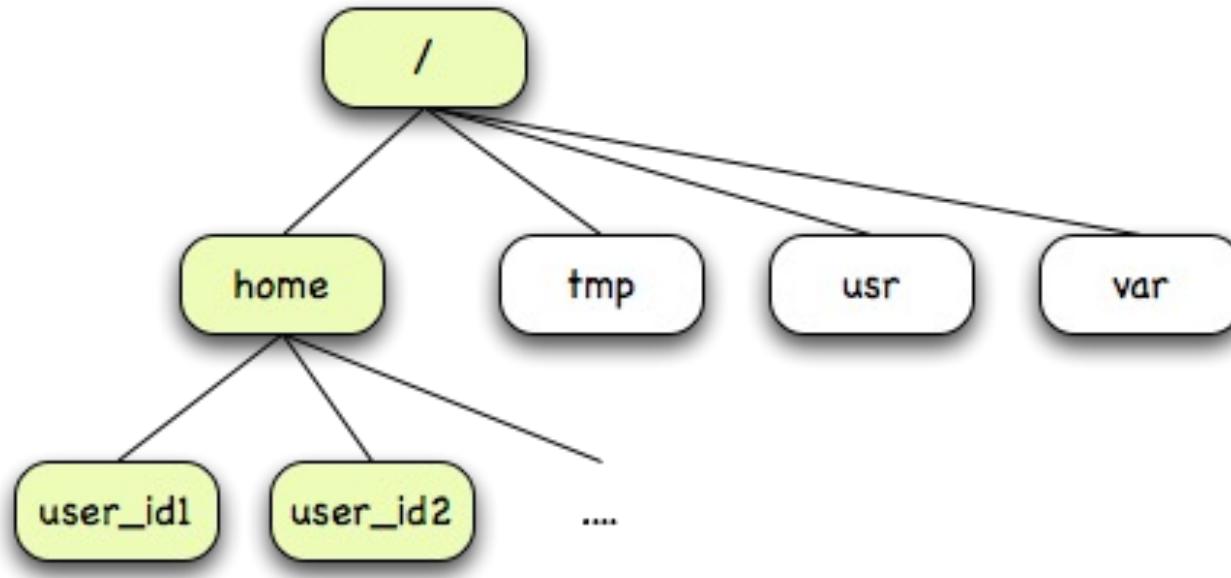
- Logging in & Changing Password
  - Login using your ID
  - Change your password immediately

```
(base) son@login0:~$ passwd
Enter login(LDAP) password:
New password:
```



# The Directory Structure

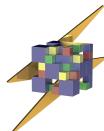
- All files are grouped together in the directory structure
- The top of the directory hierarchy is traditionally called root



# 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
- C 클러스터의 경우 a,b 클러스터와 달리 디버깅 노드 X
  - 연산이 필요한 경우 무조건 계산 노드 이용해야함



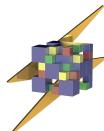
# Slurm 주요 명령어

	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 노드에서 가능
- 작업 제출 방법은 두 가지가 있음
  1. sbatch 커マン드로 작업 제출 가능. 작업 실행 커マン드와 설정을 담은 run.sh 스크립트를 준비하고, sbatch run.sh 커マン드로 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

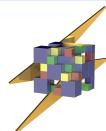
- run.sh의 내용을 아래와 같이 구성
  - single-node + multi-GPU, 아나콘다 가상환경, Python

```
#!/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

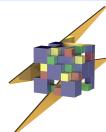
- run.sh의 내용을 아래와 같이 구성
  - single-node + single-GPU, 바이너리

```
#!/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 `youngjun@login0` → `youngjun@c00`로 변경된 것을 확인할 수 있음
- `launch-shell [#GPUs] [timelimit in minutes (optional)] [slurm partition (optional)]`
  - [#GPUs]: 0 – 3 사이의 숫자를 입력 (0 입력 시 CPU 할당)
  - [timelimit]: 1 – 720 사이의 숫자를 입력 (그 이상 숫자 입력해도 12시간 후 강제 kill)
  - Slurm partition: P1 또는 P2 입력 ( 아무것도 입력하지 않으면 P1으로 자동 배정)
- 노드 접속 후, 필요에 따라 가상환경 activate하면 됨

```
(base) youngjun@login0:~$ launch-shell 1 60
# of GPUS: 1
timelimit in minutes: 60
Slurm Partition: P1
(base) youngjun@c00:~$ █
```

```
(base) youngjun@login0:~$ launch-shell 1 60 P2
# of GPUS: 1
timelimit in minutes: 60
Slurm Partition: P2
(base) youngjun@c11:~$ █
```



# 작업 현황 확인

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

```
(base) youngjun@c11:~$ squeue
      JOBID PARTITION      NAME     USER ST      TIME  NODES NODELIST(REASON)
        37      P1    shell    kas   R      0:15      1 c00
        38      P1    shell    son   R      0:06      1 c00
        33      P2    shell youngjun R      7:44      1 c11
(base) youngjun@c11:~$ squeue -u youngjun
      JOBID PARTITION      NAME     USER ST      TIME  NODES NODELIST(REASON)
        33      P2    shell youngjun R      7:50      1 c11
(base) youngjun@c11:~$ █
```

- 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 squeue 커マン드를 통해 취소할 작업의 job id를 알아내고,  
scancel <job id> 커マン드로 작업을 취소

```
(base) youngjun@c11:~$ squeue -u youngjun
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
          33        P2    shell youngjun R      7:50      1 c11
(base) youngjun@c11:~$ scancel 33
(base) youngjun@c11:~$ srun: Job step aborted: Waiting up to 12 seconds for job step to finish.
slurmstepd: error: ** STEP 33.0 ON c11 CANCELLED AT 2022-08-10T05:57:54 **  
exit
(base) youngjun@login0:~$
```

- u 옵션으로 자신의 모든 작업을 한번에 취소 가능
  - scancel -u youngjun



# 자원 현황 확인

- sinfo (자세한 설명은 [링크](#) 참조)
  - sinfo 커맨드로 노드 및 GPU 사용 현황 확인
  - STATE 가 down인 노드가 있는 경우 관리자에게 제보 바랍니다.

```
(base) youngjun@login0:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE Nodelist
P1*          up    12:00:00      1    mix   c00
P1*          up    12:00:00     11    idle  c[01-11]
P2           up    12:00:00      5    idle  c[11-15]
```



# 5. Jupyter Notebook 사용 방법

# Jupyter Notebook 실행: launch-jupyter

- 최초 실행 시: launch-jupyter 1 10 커맨드 입력
- 최초 실행 시 Anaconda 및 Jupyter를 자동으로 설치해주며, 설치는 접속 환경에 따라 최대 30분 소요 (보통 5분 안에 완료)

```
youngjun@login0:~$ launch-jupyter 1 10
Cheking anaconda is installed... Not installed.
Installing anaconda ... Please wait.
PREFIX=/home/youngjun/anaconda
Unpacking payload ...
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/youngjun/anaconda

added / updated specs:
- _ipyw_jlab_nb_ext_conf==0.1.0=py39h06a4308_1
- _libgcc_mutex==0.1=main
- _openmp_mutex==4.5=1_gnu
- aiohttp==3.8.1=py39h7f8727e_1
- aiosignal==1.2.0=pyhd3eb1b0_0
- alabaster==0.7.12=pyhd3eb1b0_0
- anaconda-client==1.9.0=py39h06a4308_0
- anaconda-navigator==2.1.4=py39h06a4308_0
- anaconda-project==0.10.2=pyhd3eb1b0_0
- anaconda==2022.05=py39_0
- anyio==3.5.0=py39h06a4308_0
- appdirs==1.4.4=pyhd3eb1b0_0
- argon2-cffi-bindings==21.2.0=py39h7f8727e_0
- argon2-cffi==21.3.0=pyhd3eb1b0_0
```



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

- launch-jupyter [#GPUs] [conda environment (optional)] [timelimit in minutes [optional]] [slurm partition (optional)]
  - [#GPUs]: 0 – 3 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: P1 또는 P2 입력

```
(base) youngjun@login0:~$ launch-jupyter 1 base 10
Cheking anaconda is installed... OK.
Cheking jupyter is installed... OK.
# of GPUS: 1
timelimit in minutes: 10
conda env: base
Slurm Partition:
```



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

- launch-jupyter 명령을 옳게 실행하면 URL 주소가 맨 밑에 출력됨
  - 해당 URL 복사 후 127.0.0.1을 147.47.200.22로 변경하여 브라우저에 입력

```
(base) youngjun@login0:~$ launch-jupyter 1 base 10
Cheking anaconda is installed... OK.
Cheking jupyter is installed... OK.
# of GPUS: 1
timelimit in minutes: 10
conda env: base
Slurm Partition:

!!
!! Replace 127.0.0.1 with 147.47.200.22 in the url below
!!

Cheking jupyter is installed in your ENV ... OK.
[I 05:27:48.472 NotebookApp] Writing notebook server cookie secret to /home/youngjun/.local/share/jupyter/runtime/notebook_cookie_secret
[W 05:27:48.888 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[W 2022-08-10 05:27:49.388 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 2022-08-10 05:27:49.388 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 2022-08-10 05:27:49.388 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 2022-08-10 05:27:49.395 LabApp] JupyterLab extension loaded from /home/youngjun/anaconda/lib/python3.9/site-packages/jupyterlab
[I 2022-08-10 05:27:49.395 LabApp] JupyterLab application directory is /home/youngjun/anaconda/share/jupyter/lab
[I 05:27:49.400 NotebookApp] Serving notebooks from local directory: /home/youngjun
[I 05:27:49.400 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 05:27:49.400 NotebookApp] http://c00:30087/?token=68705dce6b29901bb98ba15af0201b4f65040603ce56843f
[I 05:27:49.400 NotebookApp] or http://127.0.0.1:30087/?token=68705dce6b29901bb98ba15af0201b4f65040603ce56843f
[I 05:27:49.400 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 05:27:49.405 NotebookApp]

To access the notebook, open this file in a browser:
  file:///home/youngjun/.local/share/jupyter/runtime/nbserver-4095-open.html
Or copy and paste one of these URLs:
  http://c00:30087/?token=68705dce6b29901bb98ba15af0201b4f65040603ce56843f
or http://127.0.0.1:30087/?token=68705dce6b29901bb98ba15af0201b4f65040603ce56843f
```



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

- launch-jupyter 명령을 올게 실행하면 URL 주소가 맨 밑에 출력됨
  - 해당 URL 복사 후 127.0.0.1을 147.47.200.22로 변경하여 브라우저에 입력
  - 아래와 같은 화면이 나오면 완료



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

- launch-jupyter 실행 시 유의할 점
  - launch-jupyter [#GPUs] 까지만 입력하는 것은 무방하지만, 중간의 optional argument를 생략하고 *뒤쪽의 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



# Jupyter Notebook 종료

- 단순히 브라우저 종료만으로는 노드 반납이 이뤄지지 않음!!!
- 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} 커マン드로 종료

이 버튼을 눌러  
작업을 종료!!

The screenshot shows the Jupyter Notebook dashboard. At the top, there's a navigation bar with tabs for 'Files', 'Running', and 'Clusters'. Below the navigation bar is a search bar labeled 'Select items to perform actions on them.' Underneath the search bar is a file list table. The table has columns for selection, file name, last modified date, and file size. The first item in the list is 'anaconda' (modified 3시간 전, 571 MB). The second item is 'Anaconda3-2021.05-Linux-x86\_64.sh' (modified 4달 전, 571 MB). At the top right of the dashboard, there are buttons for 'Upload', 'New', and a refresh icon. On the far right, there are 'Logout' and 'Quit' buttons, with 'Quit' being highlighted.

	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

## 6. FAQ

## 6. FAQ

- **기본적으로 서버 관련 질문은 각 연구실 별 서버 담당 인원에게 해주시기 바랍니다**
- **전체 서버관리자에게 개별 DM은 삼가주시기 바랍니다**
- 각 랩별 담당 인원은, 본인 선에서 해결하지 못할 문제의 경우 슬랙 '서버-관리' 채널에 올려주세요

교수님	담당자
Wen-Syan Li	장지수
김형신	박건도
박현우	전예지
신효필	이윤섭
오민환	박승준
이상학	조동규
이승근	안지용
이재진	손영준
이준석	김주은
차상균	채규욱
김태섭	최지인
이재윤	송종현
이홍래	박성일



## 6. FAQ

- 스토리지 용량을 늘려주세요.
  - 현재 계정별 용량 제한은 150 GB
  - 제한 상향이 필요한 경우 ppt 뒷부분 신청 양식을 참고로 내용을 작성하여 gsds-hpc1@aces.snu.ac.kr로 메일 주세요.
- 비밀번호를 초기화 해주세요.
  - 신청 양식을 참고로 내용을 작성하여 gsds-hpc1@aces.snu.ac.kr로 메일 주세요.
- 계정을 신청하고 싶어요.
  - 슬랙에 있는 신청 링크에서 계정을 신청해주세요.

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

※ 클러스터 정식 오픈 이후에는, **신청된 계정의 생성을 격주 금요일에 진행**합니다.



## 6. FAQ

- SSH가 안돼요.
  - 일반유저에게 SSH는 허용되지 않습니다. launch-shell 명령어를 통해 직접 접속을 할 수 있습니다.
- 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

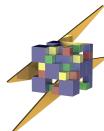
- 보안에 유의해주세요.
  - 패스워드는 별첨의 **패스워드선택및이용가이드.pdf**를 참고해주세요.
  - 패스워드를 타인 (특히 학내 구성원 외)과 공유하거나 유추하기 쉬운 것으로 설정하지 말아 주세요.

※ **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/100G메모리로 할당되어 있으므로, 더 많은 resource를 사용하고 싶은 경우 srun, salloc 등을 개인적으로 공부해서 사용하시면 됩니다.



## 7. 주의사항 (Cont'd)

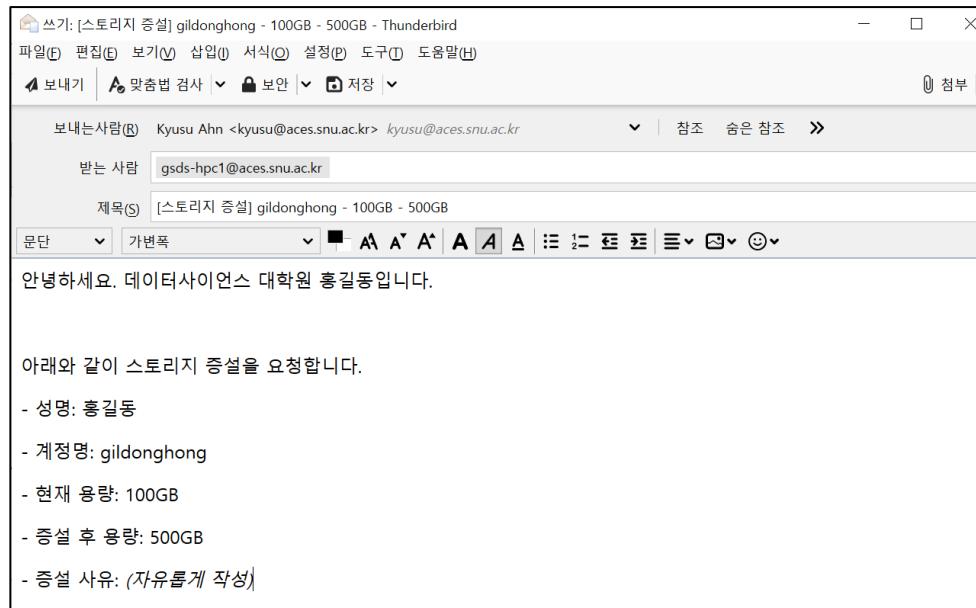
- 본인의 코드가 GPU를 사용하도록 구현되어 있지 않다면 주피터 및 shell 사용 시 (launch-jupyter, launch-shell 실행 시), [#GPUs] argument를 0으로 설정해주세요. CPU만 사용하는데 GPU를 할당 받음으로 인하여, 긴 시간동안 GPU가 필요한 다른 학우들이 GPU를 사용하지 못하는 일이 없도록 서로 배려해주세요.
  - 직접 접속의 경우 nvidia-smi를 통해 GPU usage를 확인할 수 있는데, slurm을 사용하는 GSDS cluster에서 nvidia-smi 사용 불가. GPU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



# 8. 각종 신청 양식

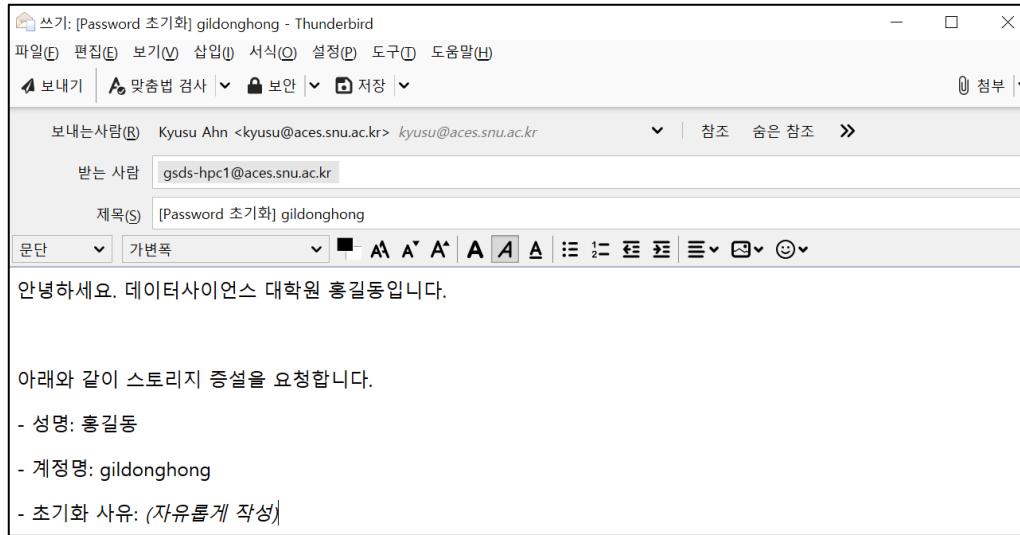
# 스토리지 증설 요청 양식

- 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

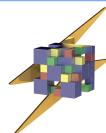
- Password 초기화를 원하는 경우, 아래의 항목을 작성하여 Cluster 관리자에게 메일 발송
  - 메일 제목: [비밀번호 초기화] {계정명}
  - 메일 내용: 성명/계정명/초기화 사유 가 모두 포함되어야 함



# 9. 매뉴얼 개정 History

## 8. 매뉴얼 개정 History

- V0.0 클러스터 세팅 후 최초 작성 (2022. 08. 10.)
  - a,b 클러스터용 매뉴얼 일부 수정



THUNDER Research Group  
Seoul National University  
서울대학교 천동 연구실

