

Seraph

▼ CLI Command

CLI Command

- **clear: 창 정리**
- **ls: 현재 디렉토리에서의 파일 리스트 출력 (L입니다)**
- **cd: 디렉토리 이동**
 - **cd {이동하고자하는 디렉토리}**
- **mkdir: 새로운 디렉토리 생성 (mkdir {디렉토리명})**
 - **현재 위치가 어딘지 확인하기**
- **mv: 파일 또는 디렉토리 이동**
 - **mv {현재파일} {이동시키고싶은 디렉토리}**
- **rm: 파일 삭제**
 - **rm {파일}: 파일 삭제**
 - **rm -f {파일}: 파일 강제 삭제**
 - **rm -r {디렉토리}: 디렉토리 삭제**
 - **rm -rf {디렉토리}: 디렉토리 강제 삭제**
- **cp: 복사**
 - **cp {복사하고자 하는 파일} {복사하고자하는 디렉토리}**
 - **cp -r {복사하고자 하는 디렉토리} {복사하고자하는 디렉토리}**

[Lecture-27_PyTorch&Seraph.pdf](#)

[Lecture-28_CNN_wSeraph.pdf](#)



권고 사항

1. server에서 debugging이 어려우니 local ipynb에서 코드 짜고 잘 돌아가는지 확인 후 서버에서 실행시키자.
2. main 함수에 코드 작성 후 if __name__ == '__main__': 에서 config를 작성하여 learning_rate, batch_size, epoch 등 hyperparameter를 수정하자

Seraph 단계별 접속

1. VSC에서 원하는 local folder 접속
2. terminal 열어서 seraph 접속
 - a. ssh guswns0429@aurora.khu.ac.kr -p 30080
 - b. password 입력

▼ 만약 최초 접속이라면?

- Anaconda를 seraph master에 설치해야한다.
- public이 아닌 private downloads쪽에 설치해야한다는 점 잊지 말자

- **Step1: Download anaconda (in Seraph master)**
 - cd /data/\$USER
 - mkdir /data/\$USER/downloads
 - cd /data/\$USER/downloads
 - wget https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Linux-x86_64.sh
- **Step2: Install Anaconda3**
 - bash ana and press TAB then Enter
 - Continue Enter until the sentence appear, "Do you accept the ~~". Then type 'yes'
 - **Anaconda3 will now be installed into this location (★★★★★)**
 - **You should type "/data/\$USERNAME/anaconda3"**
- **Step3: Init setting**
 - Do you wish the installer to initialize Anaconda3 ? → type 'yes'
 - After finishing install, type ~/.bashrc

3. seraph server측에서 내 local folder와 연결할 directory 만들고 연결
 - a. 서버측 directory 만들기
 - i. mkdir /data/guswns0429/repos/Assignment

- b. VSC에서 SFTP:config 클릭 후 개인에 맞게 설정
- c. SFTP:Download Project 클릭
 - i. 만약 오류가 발생했다면 config파일이 맞게 잘 설정됐는지 확인
- d. SFTP: Sync Local → Remote 클릭

```
{
  "name": "Aurora", #알아서 설정
  "host": "163.180.160.105", #본인이 할당받은 host 주소
  "protocol": "sftp",
  "port": 30080, #학교 안이면 22, 학교 밖이면 30080. 그냥 30080 사용하자
  "username": "guswns0429",
  "remotePath": "/data/guswns0429/repos/Assignment",
  "uploadOnSave": true,
  "useTempFile": false,
  "openSsh": false
}
```

4. server에서 모델을 돌리기 위해 필요한 data를 unzip
 - a. 마스터노드에는 데이터를 zip파일로 올리기
 - b. 본인이 사용하고자 하는 gpu에서 data.zip을 unzip해야함
 - c. gpu 접속하기

```
srunc --gres=gpu:1 --cpus-per-gpu=1 --mem-per-gpu=32G -p debug_ugrad -w aurora-g3 --pty $SHELL
```

```
srunc: AURORA: Job submitted
guswns0429@aurora-g5: /data/guswns0429/repos/CNN_Lecture$
```

접속이 완료되면 Job_submitted라고 뜨면서 해당 gpu로 접속한다.

- d. 해당 gpu에 있는 /local_datasets로 이동 후 본인이 사용할 directory 만들기

```
cd /local_datasets
mkdir guswns0429 #directory 이름을 본인 계정으로 사용하는 것이 맞다고 함
```

e. 마스터 노드에 있는 zip파일 데이터 unzip하기

```
unzip /data/guswns0429/repos/CNN_Lecture/data/cifar_10.zip
```

▼ 본인이 사용할 GPU node에서 conda init 하기

- 가상환경에서 개인별로 library를 사용해야하기 때문에 gpu에서도 마찬가지로 anaconda를 사용해야한다.

```
/data/guswns0429/anaconda3/bin/conda init  
source ~/.bashrc
```

▼ 본인이 원하는 GPU 고르기

1. slurm-gres-viz -i #이 명령어로 seraph server에 있는 gpu에 얼마나 일이 할당됐는지 확인 가능.

aurora-g1:	[GPU]	[8/8]	[#][#][#][#][#][#][#][#]	[CPU]	64/96	[MEM]	128/418.75 GiB
aurora-g2:	[GPU]	[0/8]	[-][-][-][-][-][-][-][-]	[CPU]	0/64	[MEM]	0/418.49 GiB
aurora-g3:	[GPU]	[0/8]	[-][-][-][-][-][-][-][-]	[CPU]	0/64	[MEM]	0/418.49 GiB
aurora-g4:	[GPU]	[0/8]	[-][-][-][-][-][-][-][-]	[CPU]	0/64	[MEM]	0/418.49 GiB
aurora-g5:	[GPU]	[0/8]	[-][-][-][-][-][-][-][-]	[CPU]	0/64	[MEM]	0/418.49 GiB
aurora-g6:	[GPU]	[0/8]	[-][-][-][-][-][-][-][-]	[CPU]	0/64	[MEM]	0/418.49 GiB
aurora-g7:	[GPU]	[6/6]	[#][#][#][#][#][#]	[CPU]	48/64	[MEM]	192/418.49 GiB
aurora-g8:	[GPU]	[6/8]	[-][-][#][#][#][#][#][#]	[CPU]	48/64	[MEM]	192/418.49 GiB

2. 현재 g5가 비어 있으니 이것을 사용하도록 해보자. 기억해두기

3. 해당 gpu 접속하기

5. script.sh를 통한 실행

a. script.sh를 본인의 설정에 맞게 작성

```
#!/usr/bin/bash  
  
#SBATCH -J CNN_Test  
#SBATCH --gres=gpu:1  
#SBATCH --cpus-per-gpu=8  
#SBATCH --mem-per-gpu=20G  
#SBATCH -p batch_ugrad  
#SBATCH -w aurora-g5  
#SBATCH -t 1-00:00:00  
#SBATCH -o /data/guswns0429/repos/CNN_Lecture/logs/slurm-%A_cnn_lr0.001_b512_ep10.out
```

```
pwd
which python
hostname
python3 ./code/run.py

exit 0
```

b. 가상환경에 들어가 라이브러리 설치

```
conda env list #가상환경 목록 확인
conda create -n dataviz python=3.8 -y
conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytor
ch -c nvidia #torch 다운
```

6. 실행

```
# local의 수정사항이 잘 반영되도록 sync 해주는 습관 가지기
# f1 눌러서 sync Local → Remote
sbatch script.sh
```

```
sbatch: AURORA: Job submitted
Submitted batch job 38607
```

완료되면 다음과 같이 뜸

- `squeue`를 통해 현재 작업이 진행되고 있는지 확인
 - `watch squeue`를 하면 2초 단위로 `squeue`가 자동 실행
 - `ctrl+c`하면 다시 돌아옴

7. log 결과 확인

```
cd logs
ls
cat slurm-38607_cnn_lr_0.001_b512_ep10.out
```