

H100 GPU 사용 매뉴얼 (v2.2 공개용)

워크스페이스 만들기부터 GPU 작업, 파일 전송까지

이상완

2026년 6월

전남대학교 D-AI연구실

목차

시작하기 전에

들어가서 작업하기

파일 옮기기 – 다운로드 & 업로드 (v2 신규)

파일 관리 & 공유 규칙

끝나면 꼭! 정리하기

부록 – 워크스페이스 만들기 (관리자 / 최초 1회)

자주 쓰는 리눅스 명령어

이 매뉴얼 사용법 — 먼저 읽으세요

- 이미 만들어진 워크스페이스를 쓰는 경우 (대부분의 동료)
→ **본론(들어가서 쓰기)**부터 보면 됩니다. 부록은 볼 필요 없어요.
- 워크스페이스를 처음 만드는 경우 (관리자 / 최초 1회)
→ 맨 뒤 **부록(워크스페이스 만들기)**을 먼저 보세요.

우리 연구실 사용 방식

연구실 대표 계정(lab)으로 만든 **워크스페이스 하나를 다 같이** 사용합니다.
그래서 대부분은 **만들 필요 없이** 바로 들어가서 쓰면 됩니다.

시작하기 전에

이 매뉴얼은

- 우리 연구실 H100 GPU를 쓰기 위한 AIPub 사용법입니다.
- 주소: `https://main.aipub.univ.ac.kr` (공개용 마스킹 주소)
- 흐름: 접속 → 들어가기 → 환경 준비 → GPU 작업 → 파일 옮기기 → 정리
- 명령어를 몰라도 됩니다. 화면과 복붙(복사-붙여넣기)만 따라오세요.

용어 한 줄 정리

워크스페이스 = 우리가 같이 쓰는 H100 컴퓨터 한 대 (= 컨테이너)

딱 세 가지만 기억하세요

1. 반드시 교내망에서만 사용 (집 등 학교 밖에서는 접속 안 됨)
2. 공유 계정입니다. Pause 누르기 전 다른 사람이 쓰는지 꼭 확인
3. 중요한 파일은 /lab 안에 저장 (컨테이너 안은 재시작하면 사라짐)

이 세 가지가 사고의 대부분입니다.

0단계 — 접속 (교내망 전용)

반드시 교내망에서만 사용

이 시스템은 **교내망 안에서만** 사용하도록 권고합니다.

- 교내(연구실 PC, 캠퍼스 와이파이) → 정상 접속됩니다.
- 집 등 학교 밖 → 주소를 입력해도 **접속되지 않습니다**(무한 로딩). 고장이 아니라 보안 정책입니다.
- 따라서 **교외(집 등)에서는 사용하지 마세요**. 작업은 교내에서 진행해 주세요.

0-A단계 — 보안 경고는 정상입니다 (당황 금지)

처음 접속하면 아래 같은 경고가 뜹니다. **이 사이트에 한해 정상이니** 그대로 진행하세요.

이런 화면이 떠오

“연결이 비공개로 설정되어 있지 않습니다” /
NET::ERR_CERT_AUTHORITY_INVALID

- 이유: 학내 시스템이 자체 보안 인증서를 써서 그렇습니다. 위험한 사이트가 아닙니다.
- Chrome/Edge: 고급 클릭 → “(사이트 주소)(으)로 이동” 클릭
- Safari: 세부사항 보기 → “이 웹사이트 방문” 클릭

주의: 이 통과는 해당 학내 시스템 **한 곳에서만** 하세요. 다른 사이트에서 같은 경고가 뜨면 함부로 통과하지 마세요.

들어가서 작업하기

빠른 시작 요약 (익숙해지면 이 한 장만)

교내망에서 시스템 로그인(보안경고 통과) 후:

```
1 # 1) Click workspace name -> Web Terminal
2
3 # 2) Activate env (after restart, reinstall python only)
4 apt update && apt install -y python3
5 source /lab/shared/env/bin/activate
6
7 # 3) Work in your own folder
8 cd /lab/yourname
9
10 # 4) When done: check others, then Pause
```

핵심 3가지: 교내망만 / 파일은 /lab 안에 / Pause 전 다른 사람 확인.
(yourname 자리엔 본인 폴더명 — 영문 이름 권장)

자세한 설명은 다음 장부터. 처음이면 차근차근 따라오세요.

1단계 — Web Terminal로 들어가기

1. Workspaces 목록에서 우리 워크스페이스 이름 클릭
2. 상세 화면 오른쪽 아래 Web Terminal 클릭
3. 새 창에 **까만 터미널**이 열립니다 — 여기가 H100 안입니다.

입력이 안 될 때 (자주 겪음)

- **키 입력이 안 되면** → **한/영 키를 눌러 영문 입력 상태로** 바꾸세요.
한글 입력 상태에서는 터미널에 입력이 안 됩니다.
- **커서가 안 깜빡이고 비어 있으면** → 컨테이너 준비 중일 수 있어요. 상태가 Available인지 확인하고, 창을 닫았다가 Web Terminal을 다시 누르세요.

한 번 만들어진 워크스페이스는 **어느 PC에서든 Web Terminal**로 들어갈 수 있어요(SSH 키 불필요).

2단계 — GPU 진짜 있는지 확인

터미널에 입력:

```
1 | nvidia-smi
```

아래처럼 **NVIDIA H100 80GB**가 보이면 성공입니다.

```
1 | NVIDIA-SMI 595.71.05    Driver: 595.71.05    CUDA: 13.2
2 | GPU      Name              Memory-Usage      GPU-Util
3 |  0      NVIDIA H100 80GB HBM3  69MiB/81559MiB    0%
```

80GB(81559MiB)짜리 GPU가 통째로 잡혀 있는 게 보이죠?

3단계 — 환경 켜기 (평소: 거의 매번 이것만)

이 이미지는 깡통이라, 재시작하면 **시스템 파이썬은 날아갑니다**. 하지만 PyTorch는 공용 `venv(/lab/shared/env)`에 **살아있어** 다시 설치할 필요가 없습니다.

```
1 # 1) Reinstall python only (tens of seconds, faster than torch)
2 apt update && apt install -y python3
3
4 # 2) Activate shared env (torch already inside)
5 source /lab/shared/env/bin/activate
```

확인

```
python3 -c "import torch; print(torch.cuda.is_available())"
→ True
```

프롬프트 앞에 `(env)`가 보이면 환경이 켜진 것. **torch 재설치 안 해도 됩니다**.

3단계 (참고) — 공용 venv가 없을 때만 (최초 1회)

source에서 “그런 파일 없음” 에러가 나면, 공용 환경이 아직 없는 것. **최초 1회만** 만듭니다.

```
1 apt update && apt install -y python3 python3-venv python3-pip
2 mkdir -p /lab/shared
3 python3 -m venv /lab/shared/env
4 source /lab/shared/env/bin/activate
5 pip install torch numpy          # large, takes a few min - only once
```

이건 **누군가 한 번만** 하면, 이후 모두가 위 3단계(평소)처럼 activate만 하면 됩니다. 공용 환경이니 **함부로 패키지를 지우지 마세요.**

4단계 — 토이 프로젝트: GPU 진짜 도는지 확인

테스트용 파일 `/lab/gpu_test.py`를 미리 만들어 두었습니다. 잘 실행되는지만 확인하세요.

```
1 # (env must be activated, see step 3)
2 python3 /lab/gpu_test.py
```

아래처럼 나오면 H100에서 실제로 계산이 돈 것입니다:

출력 예시

```
GPU 사용 가능: True
GPU 이름: NVIDIA H100 80GB HBM3
연산 완료! 결과 크기: torch.Size([8000, 8000])
```

축하합니다 — GPU 작업 성공!

내 코드로 바꿔 쓰려면 `cp /lab/gpu_test.py /lab/본인폴더/` 로 복사 후 수정하세요. 실제 연구 코드도 `device="cuda"`만 붙이면 GPU에서 돌립니다.

파일 옮기기 - 다운로드 & 업로드 (v2 신규)

파일 옮기기 — 길은 딱 두 갈래

파일이 지금 어디 있느냐로 방법이 갈립니다.

- 인터넷에 있는 파일 (공개 데이터셋, 깃허브 코드 등)
 - Web Terminal에서 `wget / git clone` — 키 불필요, 제일 쉬움. (주력)
- 내 노트북에만 있는 파일 (직접 만든 데이터·코드)
 - SFTP 서버로 직접 업로드 — 본인 키 필요, 유선 교내망에서만.

한 줄 요약

받기는 터미널에서(`wget/git`), 올리기는 SFTP로. 대부분의 경우는 받기만으로 충분합니다.

받기 — wget / git clone (키 불필요, 제일 쉬움)

인터넷에 있는 건 노트북을 거치지 말고 **Web Terminal**에서 바로 받으세요. 그게 제일 빠릅니다.

```
1 # (after restart) install tools
2 apt update && apt install -y wget git
3
4 # move into your folder, then download
5 mkdir -p /lab/yourname && cd /lab/yourname
6 wget https://...DOWNLOAD_URL
7 git clone https://github.com/USER/REPO.git
```

yourname=본인 폴더, DOWNLOAD_URL=받을 주소, USER/REPO=깃허브 계정/저장소.

- 공용 데이터셋은 **/lab/shared/**에 한 번만 받아 같이 쓰기 (사람마다 중복 금지)
- 노트북 파일도 **GitHub**에 올린 뒤 **clone**하는 우회가 가능
(단, 파일당 100MB 제한 / 비공개 repo는 GitHub 토큰 필요 / 민감한 데이터는 public 금지)

올리기 — SFTP, 시작 전에 수칙 3가지

내 노트북에만 있는 파일을 직접 올릴 때 씁니다. 시작 전 이것만 기억하세요.

1. 반드시 유선 교내망에서. 와이파이·개인 공유기 망에서는 Operation timed out만 뜹니다.
이건 설정 문제가 아니라 망 차단입니다. 아무리 다시 시도해도 안 됩니다.
2. 본인 공개키로 본인 SFTP 서버를 만듭니다. (워크스페이스의 키와 별개)
3. 다 쓰면 Delete. SFTP 서버는 개인 소모품 — 지워도 /lab 파일은 그대로 남습니다.

헛갈리지 마세요 — 수명 정책이 정반대

워크스페이스 = 공유물 하나. Delete 절대 금지, 끝나면 Pause.

SFTP 서버 = 개인 소모품. 필요할 때 본인 키로 만들고, 끝나면 Delete OK.

올리기 ① – SFTP 서버 만들기 (필요할 때마다, 2분)

1. 왼쪽 메뉴 SFTPServers → 오른쪽 위 + Create
2. Volume: **lab** 선택 (그래야 워크스페이스의 /lab과 같은 곳에 올라감)
3. Public key: 본인 id_rsa.pub 붙여넣기 (키가 없으면 부록 D와 같은 방법으로 생성)
4. 생성 후 상세 화면 Connection information에서 **Host / Port / User(aipub)** 확인

주의: 포트는 **만들 때마다 바뀝니다.** 외우지 말고 매번 화면에서 복사하세요.

(최초 1회) 업로드 폴더 권한 열기 – Web Terminal에서:

```
1 | mkdir -p /lab/yourname
2 | chmod 777 /lab/yourname
```

SFTP는 aipub 사용자로 들어와서 root 소유 폴더에는 못 씁니다. 이걸 안 하면 **Permission denied**가 뜹니다. (yourname=본인 폴더명)

올리기 ② — 접속해서 업로드 (유선에서!)

본인 PC 터미널에서 (호스트·포트는 SFTP 서버 상세 화면 값으로):

```
1 sftp -i ~/.ssh/id_rsa -P PORT aipub@HOST
2
3 # first time it asks to trust the host -> type "yes" in full (not "y")
4 # prompt changes to sftp> then:
5
6 lcd LOCAL_DIR      # "l" prefix = my laptop side
7 ll                 # list laptop files
8 cd /lab/yourname   # no prefix = server side
9 put FILENAME       # upload! (whole folder: put -r FOLDER)
10 bye                # quit
```

예: `sftp -i ~/.ssh/id_rsa -P 56342 aipub@203.0.113.10` (예시 IP)

핵심: 신뢰 질문엔 **yes**를 **풀로**(y 안 됨), 명령 앞 **l**이 붙으면 내 노트북 쪽 / 안 붙으면 서버 쪽.

Windows도 PowerShell에서 같은 명령 그대로 됩니다 (키 경로:

올리기 ③ — 끝나면 Delete & 막혔을 때

끝나면 정리

업로드가 끝나면 SFTP 서버는 Delete 하세요. 열린 포트를 줄이는 보안 습관이고, /lab의 파일은 안 사라집니다. 다음에 필요하면 같은 키로 또 만들면 됩니다(1분).

막혔을 때 (증상 → 처방):

- Operation timed out → **심중팔구 망 문제**. 유선 교내망인지 확인 (와이파이·개인 공유기는 차단). 명령·키 잘못이 아닙니다
- Host key verification failed → 신뢰 질문에 **yes**를 **풀로** 입력 (y는 안 됨)
- Permission denied → Web Terminal에서 `chmod 777 /lab/본인폴더` (올리기 ① 참고)
- 접속 정보가 안 맞음 → 포트는 재생성 시 바뀜. 상세 화면에서 **현재 값**을 다시 복사

파일 관리 & 공유 규칙

파일은 어디에 저장? (데이터 증발 주의)

컨테이너 안에 저장한 건 재시작하면 다 사라집니다

설치한 파이썬 /torch, /root나 아무 데나 만든 파일은 초기화됩니다.

살아남는 곳은 볼륨 /lab 한 곳뿐입니다(NAS 영구 저장, 200TB).

```
1 # OK (survives): inside the volume
2 /lab/yourname/gpu_test.py ...
3
4 # LOST (wiped on restart): inside the container
5 /root/gpu_test.py ~/ , anywhere else
```

규칙: 중요한 코드·데이터·결과는 무조건 /lab 안에(예: /lab/yourname/) 저장하세요.

공유 폴더 사용 규칙 (다 같이 쓰는 공간)

/lab 볼륨은 모두가 공유합니다. 섞이지 않게 규칙을 지켜주세요.

- 각자 본인 폴더 안에서만 작업: /lab/본인폴더/
(폴더명 규칙은 랩에서 정한 것을 따르세요. 예: 영문 이름)
- 공용 데이터셋(예: CIC-DDoS2019)은 /lab/shared/에 한 번만 받아 같이 읽기
(사람마다 중복으로 받지 않기 – 공간·시간 절약)
- 남의 폴더는 건드리지 않기

우리 랩 폴더 규칙 (2026-06 세미나 안)

- 폴더명: 영문 이름 소문자 (예: /lab/sangwan)
- 공용 데이터셋: 받기 전 단독 공지 → /lab/shared/에 저장,
shared/README.txt에 한 줄 기록
- 공용 venv: 패키지 삭제 금지, 새로 설치하면 단독 공지 (버전 충돌 시 개인
venv 권장)

공유 계정 주의 (사고 1순위)

우리는 연구실 대표 계정(lab) 하나를 다 같이 씁니다. 그래서:

- 같은 워크스페이스에 여러 명이 동시에 들어갈 수 있습니다.
- 내가 Pause/정지를 누르면 다른 사람 작업도 같이 꺼집니다.
- 파일도 공용이라, 같은 이름 파일은 서로 덮어쓸 수 있습니다.

최소 수칙

- Pause 누르기 전에 다른 사람이 쓰는 중인지 꼭 확인!
- 내 파일은 /lab/본인폴더/에 저장 (위 규칙)

우리 랩 규칙 (2026-06 세미나 안): Pause는 단독 공지 → 이의 없을 때만 / 장시간 학습은 시작 전 단독 선언(예상 소요시간 포함) + 돌리기 전 nvidia-smi로 선행 작업 확인 / 남이 학습 중일 땐 Pause 금지.

랩 운영 규칙 — 관리 사항 (2026-06 세미나 안)

- 워크스페이스가 사라졌을 때: 재생성은 관리자 1명이, 지정된 키로만.
(생성 시 SSH 키가 한 명 것만 등록되므로, 담당을 안 정하면 그날 모두 막힙니다. 공용 venv 재구축도 담당자 몫 — 3단계 참고)
- 파일 업로드(SFTP): 본인 키로 본인 서버 생성 → 유선 교내망에서 업로드 → 끝나면 Delete (“파일 옮기기” 장)
- 청소·백업: 안 쓰는 대용량 중간산출물은 학기말 정리. 중요 결과물은 각자 GitHub 등에 개인 백업 권장 (볼륨이 유일한 사본)
- 임대 만료 대비: 2026년 7월에 관리 부서에 연장/이전 확인
- 문의 창구 단일화: 관리 부서 문의는 관리자를 거쳐 1명이 (중복 문의 방지)

세미나에서 바뀐 내용은 이 장 + 앞 두 장의 규칙 블록만 고치면 됩니다.

끝나면 꼭! 정리하기

작업 끝나면 정지 (제일 중요)

워크스페이스 상세 화면 위쪽 버튼:

- Pause(정지) — 작업 잠시 멈춤. 평소엔 이걸 쓰세요.
- Delete(삭제) — 워크스페이스 자체를 없앴. 함부로 누르지 마세요(볼륨 파일은 남지만 환경은 초기화).

명심

브라우저 창만 닫는 건 정지가 아닙니다! 그리고 공유 계정이라 Pause 전에 다른 사람이 쓰는지 꼭 확인하세요.

반대로 본인이 만든 SFTP 서버는 쓰고 나면 Delete 하는 게 맞습니다(“파일 옮기기” 장 참고).

쓸 때마다 체크리스트

1. **교내망**인지 확인 (집에서는 접속 불가)
2. 시스템 접속 & 로그인 (보안 경고는 통과)
3. 워크스페이스 **이름** 클릭 → Web Terminal
4. (재시작됐으면) `apt install python3 + activate` → /lab/본인폴더에서 작업
5. **끝나면, 다른 사람 확인 후 Pause**

- 접속 자체가 안 됨 → **교내망**인지 먼저 확인 (집에서는 접속 불가)
- `nvidia-smi`가 안 됨 → 워크스페이스가 Available인지 확인
- python 없음 / `torch import` 안 됨 → `apt install python3` 후 `source ../env/bin/activate` (3단계)
- 내 파일이 사라짐 → /lab 밖에 저장했을 가능성. 다음부터 /lab/본인폴더에
- **파일 업로드(SFTP)가 안 됨** → “파일 옮기기” 장의 **올리기 ③** 참고 (**유선 교내망** 필수)
- 시스템/계정/망(방화벽) 관련 → 학내 관리 부서 (연락처는 공개용에서 마스킹)
- 그 외 / 이 문서 오류 → 작성자에게 문의

부록 — 워크스페이스 만들기 (관리자 / 최초 1회)

부록 안내 — 대부분은 볼 필요 없습니다

이 부록이 필요한 경우

워크스페이스를 처음 만드는 관리자 또는 새로 만들어야 할 때만 보세요.

- 우리 랩은 lab 계정으로 만든 워크스페이스 하나를 공유합니다.
- 이미 워크스페이스가 있으면 앞의 본론(들어가서 작업하기)만 보면 됩니다.
- 아래는 그 워크스페이스를 최초로 만드는 절차입니다.

[부록] A단계 - 생성 시작 & 프로젝트

1. 왼쪽 메뉴 Workspaces → 오른쪽 위 + Create
2. Project 칸의 가져오기 → 소속 프로젝트 선택 → OK

Workspace 이름 (규칙 있음!)

가입 아이디와 똑같이 입력하세요 (우리 경우 lab). 중복확인도 누르세요.

프로젝트 목록이 비어 있으면? 계정이 프로젝트에 아직 등록 안 된 것. 관리 부서/관리자에게 등록 요청하세요.

[부록] B단계 — 이미지 & 리소스셋 (값 주의!)

1. 이미지: 가져오기 → 프로젝트 선택 → Ubuntu 22.04 → OK
(현재 선택 가능한 이미지가 22.04뿐입니다)
2. 리소스셋: 가져오기 → 지정된 H100 리소스셋 선택 → OK
3. Resource set 개수: 정확히 50 입력

꼭 지킬 것

개수는 반드시 50. 다른 값이면 컨테이너가 생성되지 않습니다.

[부록] C단계 - 볼륨 & 컨테이너 개수

1. Volume: 가져오기 → 계정 ID와 같은 볼륨(lab) 선택 → OK
2. Mount path: /lab 입력
3. Container 개수: 1 (다른 값이면 생성 불가)

볼륨이 뭐죠?

작업 파일이 저장되는 영구 저장공간(NAS 200TB). 컨테이너가 초기화돼도 여기 데이터는 남습니다.

[부록] D단계 — SSH 공개키 (생성 시 1회 필수)

생성 화면의 Public key 칸은 필수입니다. 본인 PC 터미널에서 키를 만들어 붙여넣으세요.

Windows (명령 프롬프트 cmd):

```
1 cd %userprofile%
2 ssh-keygen -t rsa
3 type .ssh\id_rsa.pub
```

macOS / Linux (터미널):

```
1 ssh-keygen -t rsa
2 cat ~/.ssh/id_rsa.pub
```

ssh-rsa로 시작하는 전체를 복사해 Public key 칸에 붙여넣고 Create → OK. 공개키는 생성할 때만 필요하고, 이후 작업은 Web Terminal로 합니다.

같은 키를 SFTP 서버 만들 때(“파일 옮기기” 장)도 그대로 재사용합니다.

자주 쓰는 리눅스 명령어

명령어 (1) — 이동 & 둘러보기

터미널에서 “지금 내가 어디 있고 뭐가 있나”를 보는 명령어입니다.

- `ls` — 현재 폴더의 파일/폴더 목록 보기
`ls -lh` 하면 크기·날짜까지 (예: `ls -lh /lab`)
- `pwd` — 지금 내가 있는 위치(경로) 출력
- `cd` 경로 — 그 폴더로 이동 (예: `cd /lab/본인폴더`)
`cd ..` 는 한 단계 위로, `cd ~` 는 홈으로

팁

길 잃으면 `pwd`로 위치 확인 → `cd /lab/본인폴더`로 돌아오기.

명령어 (2) — 폴더 & 파일 다루기

- `mkdir` 이름 — 새 폴더 만들기 (예: `mkdir /lab/본인폴더`)
- `mv A B` — A를 B로 이동 또는 이름 변경
예: `mv gpu_test.py /lab/본인폴더/` (파일 옮기기)
- `cp A B` — A를 B로 복사 (`cp -r` 는 폴더째 복사)
- `rm` 파일 — 파일 삭제
- `rm -rf` 폴더 — 폴더 통째로 삭제 (되돌릴 수 없음!)

경고

`rm -rf`는 경로를 두 번 확인하고 쓰세요. 오타 한 번에 전부 날아갑니다. (특히 `/lab` 안에서 조심 — 공용 공간입니다.)

명령어 (3) — 확인 & 실행

- `cat` 파일 — 파일 내용 화면에 출력 (예: `cat gpu_test.py`)
- `nano` 파일 — 간단 편집기로 파일 열기/수정
저장: `Ctrl+O` → `Enter`, 종료: `Ctrl+X` (vi보다 쉬움)
- `nvidia-smi` — GPU 상태/사용량 확인
- `df -h` — 저장공간 확인 (/lab이 보임)
- `python3` 파일.py — 파이썬 코드 실행
- `pip list` — 설치된 파이썬 패키지 목록

환경 켜기 `source /lab/shared/env/bin/activate` — 작업 전 꼭. (env)가 뜨면 켜진 것.

명령어 (4) — SFTP 프롬프트 안에서 (sftp>)

파일 업로드 중 sftp> 프롬프트에서 쓰는 명령입니다. **!**이 붙으면 내 노트북, 안 붙으면 서버.

- `lpwd / lls / lcd` 경로 — **노트북** 쪽 위치 확인 / 목록 / 이동
- `pwd / ls / cd` 경로 — **서버** 쪽 위치 확인 / 목록 / 이동
- `put` 파일 — 노트북 → 서버 업로드 (`put -r` 폴더 는 폴더째)
- `get` 파일 — 서버 → 노트북 다운로드
- `bye` — 종료

팁

!명령 처럼 느낌표를 붙이면 노트북 쪽 아무 명령이나 실행됩니다 (예: `!ls -lh`).

- 작성: 이상완 / 기준일: 2026년 6월 (v2.2 공개용)
- v2 변경점: “파일 옮기기” 장 신설 – wget/git 다운로드 + SFTP 업로드 (실제 업로드까지 검증 완료)
- v2.1 변경점: 랩 운영 규칙 반영 – 폴더·shared·Pause/GPU 조율·관리 사항 (2026-06 랩 세미나 합의 기준)
- v2.2 변경점: 코드블록 한글 깨짐 수정 – 코드 안은 영문 placeholder, 한글 설명은 코드 밖으로 (복붙 안전)
- 검증 환경: AIPub (쿠버네티스 기반 학내 GPU 클라우드 플랫폼)
- 이미지: Ubuntu 22.04 / GPU: NVIDIA H100 80GB HBM3 (CUDA 13.2)
- 설치 검증: python3 + torch 2.12 + numpy (공용 venv)
- 파일 전송 검증: Web Terminal wget/git + SFTP 업로드(유선 교내망 한정 / 와이파이·개인 공유기 망 차단 확인)
- 임대 기간: ~2026년 8월 말
- 공개용 조치: 내부 주소·IP·포트·계정명·전화번호·프로젝트/리소스셋

질문은 언제든지 환영합니다