

# 한동대학교 공용서버 사용자 지침서

최종수정: 2024-08-27

---

한동대학교 빅데이터 사업단 공용서버는 교육 인프라 구축을 위한 빅데이터 저장 및 분석을 목적으로 합니다. 운영체제는 Linux OS 중 대표적인 Ubuntu (우분투) 20.04 LTS 버전을 사용하고 있습니다. 아래 절차에 따라 서버 계정 신청 및 안전한 서버 접속을 위한 SSH 설치를 완료해 주시기 바랍니다.

리눅스 시스템이 익숙하지 않은 분들은 생활 코딩의 [디렉토리](#)와 [파일](#), [셸](#)과 [커널](#) 영상과 [SSH](#) 영상을 먼저 보고 오시는 것을 권장합니다.

문의: jhmoon@handong.ac.kr

---

## 공지사항

- 2024년 1학기부터 LSF 및 NFS 내용 추가로 인해 사용 방법이 바뀌었습니다. 다음 지침서를 잘 확인하시고 진행해주세요.
- Login 서버(220.149.109.47)에서 실험을 실행하지 마세요.

# 목차

1. 한동대학교 공용 서버 시스템 구성	4
2. 계정 발급	4
2.1. 수업 목적으로 서버 사용을 원하시는 경우	4
2.2. 연구 목적으로 서버 사용을 원하시는 경우	4
3. 로그인 방법	5
3.1. 유닉스 (맥OS 또는 리눅스) 환경에서 공용 서버 접속하기	5
3.2. 윈도우 환경에서 접속하기	5
3.2.1. "MobaXterm"으로 접속하기	5
3.2.2. "putty" 로 접속하기	5
4. 사용자 비밀번호 변경	6
5. 작업디렉터리 및 쿼터 정책	6
6. 파일 송수신하기	7
6.1. 리눅스/유닉스 계열	7
6.2. 윈도우	7
6.3. 웹으로부터 파일 받기	8
7. 서버 사용하기	8
7.1. Job 할당 받기	8
7.1.1. LSF 명령어	9
7.1.2. Batch Job 할당	12
7.1.3. Interactive Job 할당	12
7.1.4. 할당이 안될 경우	13
7.1.5. 현재 실행 중인 서버 확인	13
7.1.6. Job 종료	13
7.2. conda 사용법	13
7.2.1. conda 설치하기	13
7.2.2. conda 환경 확인하기	14
7.2.3. conda 개인 환경 생성	14
7.2.4. conda 환경 활성화 시키기, 필요한 패키지 추가 설치	15
7.2.5. conda 환경 비활성화 시키기	15
7.2.6. conda 환경 삭제하기	15
7.3. Jupyter Notebook 사용법	15
7.3.1. LSF 를 통해 job 할당 받기	15
7.3.2. Jupyter Notebook 실행	15
7.3.3. SSH Tunneling	16
7.3.4. Jupyter Notebook 접속	16
7.3.5. Jupyter Notebook 종료	16
7.4. podman 사용	16
7.4.1. LSF 를 통해 job 할당 받기	16
7.4.2. MySQL Server 실행	17
7.4.3. SSH Tunneling 진행	17

7.4.4. MySQL 접속	18
7.4.5. Podman 종료	18
8. 기타 리눅스 명령어들	18
9. 문의 및 공용 프로그램 / 라이브러리 설치 신청	19

# 1. 한동대학교 공용 서버 시스템 구성

공용서버 한대의 시스템 사양은 아래와 같으며, 총 두 대의 서버와 한 대의 로그인 서버가 있습니다. 사용자는 로그인 서버에서 두 서버 중 하나를 랜덤하게 할당받아 사용 가능합니다.

	모델	개수 / 용량
CPU	Intel Xeon Silver 4314	2
GPU	A30	4
RAM	DDR4-3200	512GB
SSD		17TB


\* 1개의 물리적인 GPU는 Multi Instance GPUs 기술이 적용되어 최대 4개의 논리 인스턴스로 분할되어 있습니다. 필요량에 따라 인스턴스의 크기를 선택하여 할당받으실 수 있습니다.  
물리 GPU 1개 memory: 24576MB, 4분할 된 논리 인스턴스 1개의 memory: 5952MB.

## 2. 계정 발급

- 계정 발급 시간: 평일 오후 5시 ~ 7시. 최대 1주일 소요.
- 메일이 오지 않는다면 스팸메일함을 확인 후 관리자에게 문의해주세요.

### 2.1. 수업 목적으로 서버 사용을 원하시는 경우

- 1) 수업 수강생들의 계정을 한번에 신청  
아래 링크의 엑셀 시트를 다운로드 받은 뒤 계정이 필요한 수강생 정보들을 작성하여 관리자 메일로 보내시면 한번에 계정을 생성할 수 있습니다.

 한동대학교 공용서버 계정 신청서\_엑셀버전

### 2.2. 연구 목적으로 서버 사용을 원하시는 경우

- 1) 아래 링크의 계정 신청서를 작성하여 제출해주세요  
<https://forms.gle/HKjfMwP4Z9XcnnhK9>  
계정이 만들어진 뒤에 이후의 과정을 진행 할 수 있으며, 계정이 생성되면 제출하신 메일로 계정과 초기 패스워드를 알려드립니다.  
\* 서버 관련 사항은 모두 작성하신 학교 메일 주소로 전달되며, 학교 계정이 끝난 경우 계정이 제거됩니다.

### 3. 로그인 방법

빅데이터 사업단 공용서버가 속한 한동대학교 네트워크는 보안상 망 외부로부터의 접속을 차단하고 있습니다. 학교 내부 네트워크에서만 공용 서버에 접근이 가능합니다.

IP 주소	
Host명	hgucoss
ip 주소	220.149.109.47

#### 3.1. 유닉스 (맥OS 또는 리눅스) 환경에서 공용 서버 접속하기

[영상 Tutorial](#)

우분투 desktop 버전 또는 맥 OS는 기본적으로 `ssh client`가 설치되어 있습니다. `terminal`을 열어 아래 명령어로 서버에 접속합니다.

```
$ ssh <사용자 ID>@220.149.109.47
<사용자 ID>@220.149.109.47's password:
```

계정과 초기 비밀번호, 할당된 서버는 제출한 메일로 공지 됩니다.  
로그인 시 비밀번호를 입력해도 화면에 표시되지 않습니다.

#### 3.2. 윈도우 환경에서 접속하기

[영상 Tutorial](#)

[MobaXterm](#) 또는 [putty](#) 등 `ssh client` 프로그램을 이용하여 접속 가능합니다. 위 프로그램들은 인터넷에서 무료로 다운로드 받아 사용하실 수 있습니다.

##### 3.2.1. "MobaXterm"으로 접속하기

MobaXterm 설치 후 실행.

왼쪽 위 `Session` 탭 클릭, `SSH` 선택

`Remote host`에 서버ip주소(220.149.109.47) 입력

(옵션)`Specify username`: 발급 받은 계정 입력, `OK` 누르기

왼쪽 창에 세션이 저장되었으며, 오른쪽 창에 로그인 화면 확인. 패스워드 입력  
서버 접속 완료.

##### 3.2.2. "putty" 로 접속하기

putty 받은 후 실행

`Hostname(or IP address)`: 220.149.109.47 입력

Port: 22  
Connection type: SSH

Open 버튼 클릭  
ID와 패스워드 입력하여 접속

Using username "<사용자 ID>".  
<사용자 ID>@220.149.109.47's password:

## 4. 사용자 비밀번호 변경

초기 비밀번호로 접속 후 `passwd` 명령어로 패스워드를 변경해 주세요

```
$ passwd
Changing password for "<사용자 ID>"
Current password: //현재 패스워드 입력
New password: //변경할 패스워드 입력
Retype new password: //변경할 패스워드 한번 더 입력
```

비밀번호를 입력해도 화면에 표시되지 않습니다.

새로운 패스워드는 아래 정책을 따라 만들어주세요.

패스워드 보안정책

- [1] 사용자 패스워드 길이는 최소 9자이며, 영문 대/소문자, 숫자, 특수문자의 조합으로 이뤄져야 한다. 영문 사전 단어는 사용이 불가하다.
- [2] 사용자 패스워드 변경 기간은 2개월로 설정(60일) 된다.
- [3] 최대 로그인 실패 허용 횟수 : 5회

- 5회 이상 틀릴 경우, 해당 계정은 lock이 걸리므로, 운영자에게 문의해야 한다.
- 같은 PC에서 접속을 시도하여 5회 이상 틀릴 경우, 해당 PC의 IP 주소는 일시적으로 block 되므로 이 경우에도 운영자에게 문의해야 한다.

## 5. 작업디렉터리 및 쿼터 정책

계정당 사용 가능한 용량과 파일 수 제한은 아래와 같습니다.

디렉터리 경로	용량제한
/home/<사용자 ID>	100GB

추가적인 용량이 필요하신 경우 관리자에게 연락 부탁드립니다.

## 6. 파일 송수신하기

### 6.1. 리눅스/유닉스 계열

[영상 Tutorial](#)

scp를 사용해서 개인 컴퓨터에서 서버로 파일/디렉터리 옮기기

```
$ scp '클라이언트의 파일 경로' <사용자 ID>@서버주소:'서버의 파일 경로'
$ scp -r '클라이언트의 디렉터리 경로' <사용자 ID>@서버주소:'서버의 디렉터리 경로'
```

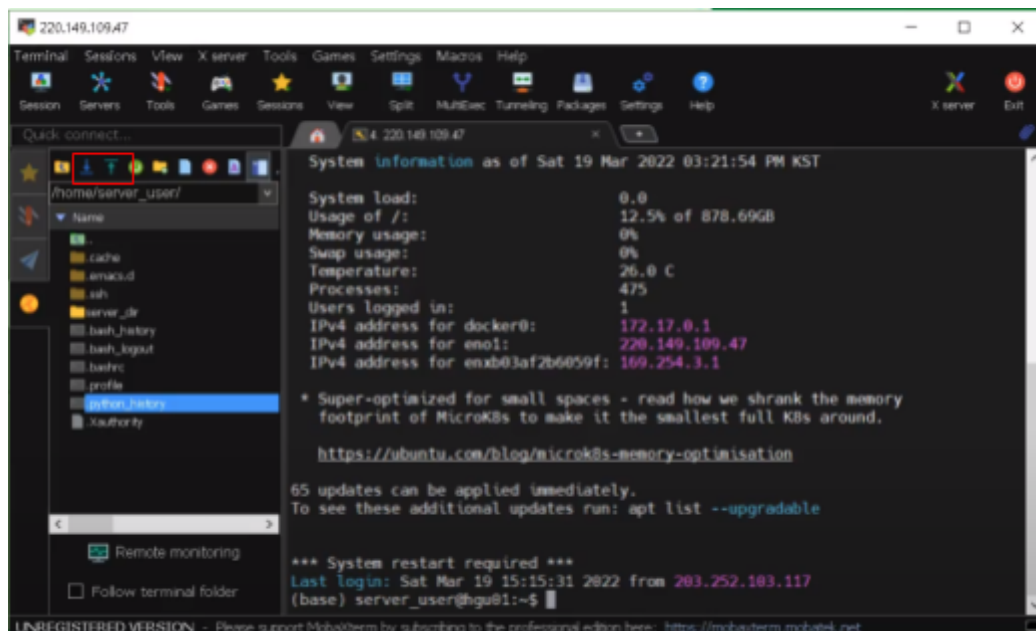
서버에서 개인 컴퓨터로 파일/디렉터리 옮기기

```
$ scp <사용자 ID>@서버주소:'서버의 파일 경로' '클라이언트의 파일 경로'
$ scp -r <사용자 ID>@서버주소:'서버의 디렉터리 경로' '클라이언트의 디렉터리 경로'
```

### 6.2. 윈도우

[영상 Tutorial](#)

[MobaXterm](#) 의 sftp로 파일/디렉토리 옮기기



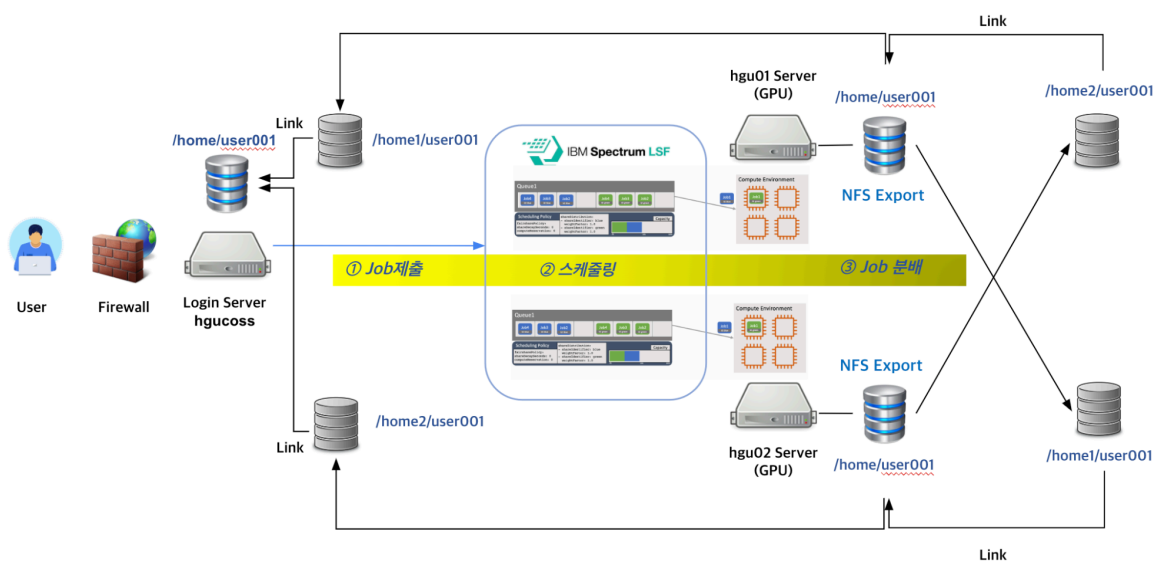
서버 접속 후 업로드, 다운로드 아이콘(빨간색 박스)을 이용하여 파일 옮기기 가능

## 6.3. 웹으로부터 파일 받기

wget, curl, aria2c을 사용해서 url로부터 파일을 웹에서 공용서버로 바로 받을 수 있습니다.

```
$ wget <다운로드 받을 url> <파일을 받을 폴더 경로>
-r : 디렉토리 구조와 파일을 그대로 하여 복사
-b : 백그라운드에서 파일다운로드 실행
-c : 연결이 끊긴 시점부터 이어서 파일 다운로드
```

## 7. 서버 사용하기



공용 서버를 사용하기 위해서는 Login 서버(hgucoss)에서 현재 사용이 가능한 Computer 서버(hgu01 혹은 hgu02)를 할당 받아야 합니다.

- Login 서버에서 job을 할당 받을 때, 필요한 자원 및 사용 시간을 같이 전달해줘야 합니다.
  - 작성한 자원 혹은 시간 이상을 사용하게 되면 할당된 서버가 중단됩니다.
  - 요청한 자원이 많거나 실행 중인 process가 많은 경우, 할당받는 시간이 많이 소요됩니다.
- Login 서버에서 실험을 진행하시면 안됩니다.

### 7.1. Job 할당 받기

Login 서버를 통해 공용 서버를 할당 받게 되면 각 계정별로 컨테이너를 할당받아 개발환경을 구축할 수 있습니다. 각 사용자는 자신의 컨테이너 내부에서는 관리자 권한을 갖고 있어 자유롭게 필요한 프로그램을 설치할 수 있습니다.

- LSF 명령어 및 Job 할당은 Login 서버에서만 진행이 됩니다.



- 할당 받은 **job의 id (username)** 을 기억하면 좋습니다.

### 7.1.1. LSF 명령어

- Cluster: LSF 로 구성되어 있고 LSF 작업을 수행하는 모든 host들의 집합
  - bhosts: cluster 내의 hosts 정보 보기

```
[jhmoo@hguccoss:~$ bhosts
HOST_NAME      STATUS      JL/U      MAX  NJOBS      RUN  SSUSP  USUSP  RSV
hgu01          ok          -         32    0         0     0     0     0
hgu02          closed     -         16   16        16     0     0     0
hguccoss       closed     -          0    0         0     0     0     0
```

- lsid: cluster 이름 보기

```
[jhmoo@hguccoss:~$ lsid
IBM Spectrum LSF Standard 10.1.0.14, Jan 08 2024
Copyright International Business Machines Corp. 1992, 2016.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

My cluster name is hgu
My master name is hguccoss
```

- lsclusters: cluster 의 상태와 size 보기

```
[jhmoo@hguccoss:~$ lsclusters
CLUSTER_NAME  STATUS  MASTER_HOST  ADMIN  HOSTS  SERVERS
hgu           ok      hguccoss     lsadmin  3      3
```

- Job: 실행을 위해 LSF 에 submit 되어지는 단위

- bjobs: cluster 내의 job 정보 보기

실행 중인 Job의 ID를 까먹은 경우, bjobs을 통해 확인 가능

- bjobs -u [username] : 특정 사용자의 job을 보고자 할 때 사용
- bjobs -p : pending reasons 을 보여줌
- bjobs -l [job ID] : job에 대한 자세한 정보를 보여줌  
실행하는 데 사용된 메모리, 시간 등을 볼 수 있습니다.

```
[jhmoo@hguccoss:~$ bjobs -l 11393

Job <11393>, Job Name <train>, User <jychun>, Project <default>, Status <RUN>,
Queue <class>, Interactive pseudo-terminal shell mode, Command </bin/bash>
Thu Aug 22 20:41:16: Submitted from host <hguccoss>, CWD <$HOME/projects/jump-ai-2024-test>, 16 Task(s), Requested Resources <rusage[mem=16384.00]>;
Thu Aug 22 20:41:16: Started 16 Task(s) on Host(s) <16*hgu02>, Allocated 16 Slot(s) on Host(s) <16*hgu02>;
Tue Aug 27 21:03:36: Resource usage collected.
The CPU time used is 13126528 seconds.
MEM: 5.7 Gbytes; SWAP: 0 Mbytes; NTHREAD: 1147
PGID: 601885; PIDs: 601885
PGID: 601887; PIDs: 601887
PGID: 601889; PIDs: 601889
PGID: 1453882; PIDs: 1453882 1476272
PGID: 1476315; PIDs: 1476315 1476404 1476436 1476437
1476438 1476439 1476440 1476441 1476442 1476443 1476444
1476445 1476446 1476447 1476448 1476449 1476450 1476451
1476452 1476453 1476454 1476455 1476456 1476457 1476458
1476459 1476460 1476461 1476462 1476463 1476464 1476465
1476466 1476467 1476468

MEMLIMIT
16 G

MEMORY USAGE:
MAX MEM: 22 Gbytes; AVG MEM: 4.7 Gbytes; MEM Efficiency: 138.04%
```

- `bjobs -r` : run 되고 있는 job만 보여줌
- `bjobs -q [queue name]` : 특정 queue에 있는 job을 보여줌

```

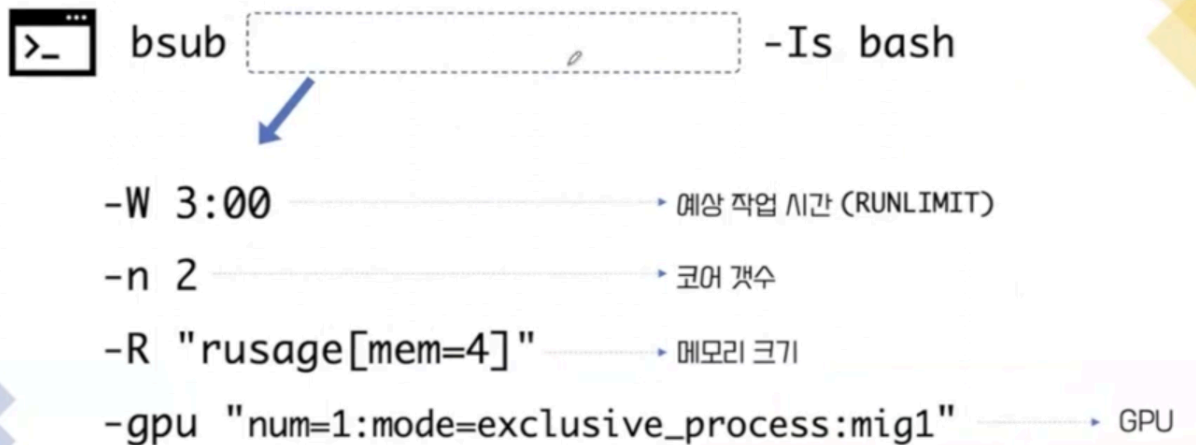
[jhmoon@hgucoss:~]$ bjobs -u all
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
11393  jychun  RUN   class  hgucoss    16*hgu02  train     Aug 22 20:41

```

#### ○ **bsub: job submit** 하기

- `-B` : job이 dispatch 되거나 run 될 경우 사용자에게 mail을 보냄
- `-N` : job이 끝날 때 mail을 보냄
- `-I [-ls]` : Interactive Job 실행 [7.1.3 확인]
- `-K` : job이 끝날 때까지 기다림 (`-l` option과 같이 사용할 수 없음) [7.1.2 확인]
- `-gpu "num=num_gpus:mode=shared | exclusive_process:mig=num_mig"`
  - `num_gpus` : 할당하고자 하는 GPU 수량 작성
  - `mode` : Shared or Exclusive mode 지정
  - `num_mig` : MIG(Mlti Instance GPU) 수량 작성
- `-o output_file` : standard output file 지정
  - batch job 을 실행하고 난 후의 작업 결과물을 저장하고자 할 때 사용
- `-e error_file` : standard error file 지정
- `-J "job_name"` : Job name 지정, 원하는 이름 설정 가능
- `-m machine_name` : Job을 지정한 system들에서 실행
- `-n` : cpu 개수를 지정
  - 지정하지 않을 경우 cpu는 1개만 할당됩니다.
- `-q queue_name` : 특정 queue로 job을 보냄
  - 총 8개의 queue 가 존재합니다.
    - `class`: 수업을 위한 용도로 TA 혹은 해당 교수님만 사용 가능
    - `maze`: 빅데이터 모델링과 플랫폼 수업을 위한 용도로 TA 혹은 해당 교수님만 사용 가능
    - `jupyter`: jupyter notebook을 실행할 때 사용
    - `jupyter_gpu`: jupyter notebook으로 실험을 하는데 gpu가 필요할 때 사용
    - `container`: docker 와 같은 실험이 필요할 때 사용
    - `container_gpu`: docker 와 같은 실험을 하는데 gpu가 필요할 때 사용

- normal: 지정하지 않을 경우 할당
- batch
- -R "rusage[mem=#]": resource 를 요구할 때 사용
  - 지정하지 않을 경우 **memory**는 **1G**만 할당됩니다.
  - # 에는 할당하고자 하는 **memory [MB] 크기**를 작성해주시면 됩니다.
- -W : run time limit
  - 지정하지 않을 경우 **1시간** 할당되며, 그 시간 이후 할당된 job은 종료됩니다.
    - 지정한 queue에 따라 최대 할당 가능 시간이 다릅니다.
      - normal : 최소 1시간 ~ 최대 24시간 지정 가능
      - jupyter : 최소 3시간 ~ 최대 24시간 지정 가능
      - jupyter\_gpu : 최소 3시간 ~ 최대 24시간 지정 가능
      - container : 최소 3시간 ~ 최대 24시간 지정 가능
      - container\_gpu : 최소 3시간 ~ 최대 24시간 지정 가능
      - batch : 최소 1시간 ~ 최대 24시간 지정 가능



- bqueues: Queue의 우선순위, 상태 및 job 정보 표시

```
[jhmoon@hgucoss:~$ bqueues
```

QUEUE_NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP
class	70	Open:Active	-	-	-	-	16	0	16	0
batch	50	Open:Active	-	-	-	-	0	0	0	0
jupyter	50	Open:Active	-	-	-	-	0	0	0	0
jupyter_gpu	50	Open:Active	-	-	-	-	0	0	0	0
container	50	Open:Active	-	-	-	-	0	0	0	0
container_gpu	50	Open:Active	-	-	-	-	0	0	0	0
maze	50	Open:Active	-	-	-	-	0	0	0	0
normal	30	Open:Active	-	-	-	-	0	0	0	0

- **bkill [Job ID] : Job 종료하기**

- 진행 중인 job을 끝 수 있습니다.

```
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
3835   jychun  RUN   normal  hgucoss    hgu02      *mycode.py Apr 18 21:36
(base) jychun@hgucoss:~/projects/lsf-test$ bkill 3835
Job <3835> is being terminated
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
No unfinished job found
```

### 7.1.2. Batch Job 할당

Batch Job이란 사용자의 개입 없이 정기적으로 실행되는 예약된 백그라운드 프로그램입니다. 일반적으로 장기 메모리를 소비하는 대량의 데이터를 처리하고, 사용자와 상호작용이 덜 필요한 프로그램을 실행하는 데 사용됩니다.

- Batch Job의 예로는 Java, Bash 등이 존재합니다.

작업이름                      출력물 저장될 파일

```
>_ bsub -J "test" -o output.txt python long.py
```

나의 작업

```
(base) jychun@hgucoss:~/projects/lsf-test$ bsub python mycode.py
Job <3833> is submitted to default queue <normal>.
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
3833   jychun  RUN   normal  hgucoss    hgu02      *mycode.py Apr 18 21:32
```

- 다음과 같이, 사용자의 개입 없이 코드만 실행하고 싶은 경우에 해당
  - `bsub [program] [file name]` 을 통해, 최소한의 자원으로 코드를 실행할 수 있도록 job 을 할당할 수 있습니다.

### 7.1.3. Interactive Job 할당

Interactive Job은 사용자가 시스템과 실시간으로 상호작용하며 수행하는 작업을 의미합니다. 작업이 수행되는 동안 사용자가 직접 명령을 입력하거나, 결과를 실시간으로 확인하고 그에 따라 추가 명령을 실행하는 형태의 작업에 해당됩니다.

- Interactive Job의 예로는 Jupyter Notebook 등이 존재합니다.

Interactive Job을 할당받기 위해서는 `bsub` 뒤 **-Is** 를 넣어주셔야 합니다.

```
>_ bsub -J "test" -Is bash
```

Interactive

```
(base) jychun@hgucoss:~/projects/lsf-test$ bsub -J "ijob" -Is bash
Job <3841> is submitted to default queue <normal>.
<<Waiting for dispatch ...>>
<<Starting on hgu02>>
(base) jychun@hgu02:~/projects/lsf-test$ python mycode.py
```

- 위 그림의 경우 할당 받은 'hgu02' 서버에서 실험 등을 진행하시면 됩니다.
- screen을 사용하지 않는 상황에서 exit 을 진행하게 되면 실행 중인 job이 종료됩니다.

#### 7.1.4. 할당이 안될 경우

- bjobs 를 통해 STAT 확인
  - 다음과 같이 STAT 가 PEND 인 경우, 요청한 resource 를 할당하기 위해 기다리는 중입니다. 다른 사용자가 종료함으로써, 요청한 resource 만큼 자리가 나면 순서대로 job이 진행됩니다.

```
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
3834   jychun  PEND  normal  hgucoss              *mycode.py Apr 18 21:34
```

#### 7.1.5. 현재 실행 중인 서버 확인

Login 서버에서는 어떠한 실험도 진행이 되어서는 안됩니다. 실험을 진행하시면서 현재 서버의 위치를 잘 확인하시기 바랍니다.

- hostname 명령어를 통해, 현재 서버의 위치를 확인해주세요

```
(base) jychun@hgucoss:~/projects/lsf-test$ hostname
hgucoss
(base) jychun@hgucoss:~/projects/lsf-test$
(base) jychun@hgucoss:~/projects/lsf-test$
(base) jychun@hgucoss:~/projects/lsf-test$ bsub -J "ijob" -Is bash
Job <3842> is submitted to default queue <normal>.
<<Waiting for dispatch ...>>
<<Starting on hgu01>>
(base) jychun@hgu01:~/projects/lsf-test$ hostname
hgu01
```

#### 7.1.6. Job 종료

bkill [job ID] 를 통해 실행하는 job 을 종료해주세요.

```
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
3835   jychun  RUN   normal  hgucoss    hgu02      *mycode.py Apr 18 21:36
(base) jychun@hgucoss:~/projects/lsf-test$ bkill 3835
Job <3835> is being terminated
(base) jychun@hgucoss:~/projects/lsf-test$ bjobs
No unfinished job found
```

## 7.2. conda 사용법

공용서버의 기본 컨테이너 이미지에는 [miniconda](#)가 설치되어 있지 않습니다. 아래를 참고하여 필요한 환경을 구성하여 사용해주세요

### 7.2.1. conda 설치하기

서버를 할당 받은 뒤, 다음과 같이 진행해주세요.

1. wget  
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh
2. bash Miniconda3-latest-Linux-x86\_64.sh
3. 'enter' 누르기
4. Do you accept the license terms? [yes | no] 가 나오면 'yes'

```
llowing conditions are met:

* Redistributions of source code must retain

Do you accept the license terms? [yes|no]
[no] >>>
```

5. 'enter'

```
/home/jhmoon/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/jhmoon/miniconda3] >>>
```

6. 'yes'

```
You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[[no] >>> yes
```

7. 할당된 서버를 'exit' 해서 나간 후 다시 접속 시, 설치 완료

```
(base) jhmoon@hgu01:~$ conda env list
# conda environments:
#
base * /home/jhmoon/miniconda3
```

### 7.2.2. conda 환경 확인하기

아래 명령어를 통해 설치 된 환경을 확인할 수 있으며, 기본 환경은 다음과 같습니다.

```
$ conda env list
# conda environments:
#
base * /usr/local/miniconda3
```

### 7.2.3. conda 개인 환경 생성

자신에게 필요한 환경을 새로 생성합니다.

```
$ conda create -n <환경 이름> python=3.8 # 설치하고자 하는 python 버전 선택
$ conda create -n <환경 이름> python=3.8 numpy torch # 필요한 패키지도 함께 설치 가능합니다.
```

### 7.2.4. conda 환경 활성화 시키기, 필요한 패키지 추가 설치

```
$ conda activate <환경 이름> # 자신이 필요한 환경을 활성화 시킵니다.
$ conda install <packages> # 방법1. conda install을 사용하여 패키지 설치.
$ pip install <packages> # 방법2. pip install을 사용하여 패키지 설치.
```

```
$ conda create -n <새 환경 이름> -clone <기존 환경 이름>
```

### 7.2.5. conda 환경 비활성화 시키기

```
$ conda deactivate # 현재 활성화 된 환경을 비활성화 시킵니다.
```

### 7.2.6. conda 환경 삭제하기

```
$ conda remove -n <환경 이름> # 선택한 환경을 삭제합니다.
```

## 7.3. Jupyter Notebook 사용법

### 7.3.1. LSF 를 통해 job 할당 받기

할당 받은 job ID 와 compute node 를 기억해주세요.

```
bsub -Is -q jupyter bash
```

```
Job <3724> is submitted to queue <jupyter>.
<<Waiting for dispatch ...>>
<<Starting on hgu01>>
```



### 7.3.2. Jupyter Notebook 실행

위 7.2 conda 사용법 을 따라 conda 를 설치한 후, 진행해 주시기 바랍니다.

- 이때, port 는 관리자로부터 할당 받은 port 번호를 사용해주시기 바랍니다.

```
screen -S notebook
conda activate dev
jupyter notebook --no-browser --ip=0.0.0.0 --port=55316
```

### 7.3.3. SSH Tunneling

Jupyter notebook 을 실행하면서 받은 URL 은 잠시 메모장에 적어두고 다음을 진행해주세요.

Compute node인 hgu01과 hgu02 서버는 hgucoss 를 통해 접속이 가능하기 때문에, port mapping 구간이 필요합니다.

- ssh tunneling 이 진행 중인 terminal 을 종료하게 되면, 접속이 종료됩니다.

ssh tunneling 을 실행 하기 위해서는 다음을 따라 주세요.

1. screen 을 사용 중인 경우, 'CTRL+AD' 를 눌러 screen 을 나와주세요.
2. 할당 받은 Compute node 를 나와주세요. 이때, 'exit' 명령어를 사용하시면 됩니다.
3. 현재 위치가 hgucoss 인 것을 확인 후, ssh tunneling 을 실행해주세요.

**ssh -L :[port 번호]:localhost:[port 번호] -N [account\_name]@[할당 받은 compute node]**

- port 번호는 관리자로부터 받은 번호를 사용해주세요.
- 할당 받은 compute node 는 (LSF로 할당 받았던) hgu01 혹은 hgu02 를 넣어주시면 됩니다.
- 작업이 진행 중에는 ssh tunneling 을 종료하지 말아주세요.

```
(base) jychun@hgucoss:~$ ssh -L :58888:localhost:58888 -N jychun@hgu02
jychun@hgu02's password:
```

### 7.3.4. Jupyter Notebook 접속

7.3.2 에서 할당 받은 URL 에 접속을 진행하시면 됩니다.

**http://220.149.109.47:[port번호]/?token=**

### 7.3.5. Jupyter Notebook 종료

1. ssh tunneling 을 실행했던 terminal 을 종료합니다.
2. job을 할당 받았던 compute node 로 돌아가서  
CTRL+C 를 통해 실행 중인 Jupyter notebook 을 종료합니다.



+ screen 을 실행 중인 경우, screen -r [이름] 을 통해 이동 후 진행해주세요.

\* 할당 받았던 compute node 로 돌아가기 위해서는

ssh [hgu01 | hgu02] 를 사용하시면 됩니다.

3. 할당 받았던 compute node 에서 'exit' 을 통해 1차 서버를 종료합니다.

4. hgucoss 서버에서 할당 받은 job 이 종료되었는지 2차 확인을 합니다.

종료가 되지 않았다면, bkill 을 통해 서버를 종료해주시기 바랍니다.

## 7.4. podman 사용

### 7.4.1. LSF 를 통해 job 할당 받기

할당 받은 job ID 와 compute node 를 기억해주세요.

```
bsub -Is -q jupyter bash
```

```
Job <3724> is submitted to queue <jupyter>.  
<<Waiting for dispatch ...>>  
<<Starting on hgu01>>
```

### 7.4.2. MySQL Server 실행

사용하고자 하는 podman image 를 pull 진행해주세요.

- 이때, 오류가 발생하는 경우 'podman system migrate' 을 진행해주세요.
  - 그럼에도 오류가 발생하는 경우, 관리자에게 메일 문의 부탁드립니다.

```
podman pull docker.io/library/mysql:8.0.32
```

사용하고자 하는 image 를 pull 한 뒤 podman 을 실행해주시면 됩니다.

- 이때, port 는 관리자로부터 할당 받은 port 번호를 사용해주시기 바랍니다.

```
podman run -d --name aix_mysql \  
--rm \  
-p 3306:3306 \  
-v $(pwd)/data:/var/lib/mysql \  
-v $(pwd)/scratch:/scratch \  
-e MYSQL_ROOT_PASSWORD=abcd1234 \  
docker.io/library/mysql:8.0.32
```

### 7.4.3. SSH Tunneling 진행

Compute node인 hgu01과 hgu02 서버는 hgucoss 를 통해 접속이 가능하기 때문에, port mapping 구간이 필요합니다.

- ssh tunneling 이 진행 중인 terminal 을 종료하게 되면, 접속이 종료됩니다.

ssh tunneling 을 실행 하기 위해서는 다음을 따라 주세요.

4. screen 을 사용 중인 경우, 'CTRL+AD' 를 눌러 screen 을 나와주세요.
5. 할당 받은 Compute node 를 나와주세요. 이때, 'exit' 명령어를 사용하시면 됩니다.
6. 현재 위치가 hgucoss 인 것을 확인 후, ssh tunneling 을 실행해주세요.

**ssh -L :[port 번호]:localhost:[port 번호] -N [account\_name]@[할당 받은 compute node]**

- port 번호는 관리자로부터 받은 번호를 사용해주세요.
- 할당 받은 compute node 는 (LSF로 할당 받았던) hgu01 혹은 hgu02 를 넣어주시면 됩니다.
- 작업이 진행 중에는 ssh tunneling 을 종료하지 말아주세요.

```
(base) jychun@hgucoss:~$ ssh -L :58888:localhost:58888 -N jychun@hgu02
jychun@hgu02's password:
```

### 7.4.4. MySQL 접속

DataGrip 등을 사용하여 접속을 해야 하는 경우,

- Host: 220.149.109.47
- Port: SSH tunneling 에서 사용했던 port 번호
- Username: 할당 받은 account\_name
- Password: 할당 받은 password

### 7.4.5. Podman 종료

1. ssh tunneling 을 실행했던 terminal 을 종료합니다.
2. job을 할당 받았던 compute node 로 돌아가서  
podman stop [container\_name] 을 통해 실행을 종료합니다.  
\* 할당 받았던 compute node 로 돌아가기 위해서는  
**ssh [hgu01 | hgu02]** 를 사용하시면 됩니다.
3. 할당 받았던 compute node 에서 'exit' 을 통해 1차 서버를 종료합니다.
4. hgucoss 서버에서 할당 받은 job 이 종료되었는지 2차 확인을 합니다.  
종료가 되지 않았다면, bkill 을 통해 서버를 종료해주시기 바랍니다.

## 8. 기타 리눅스 명령어들

```
$ top : 실행중인 프로세스 상태 확인하기
$ htop : 실행중인 프로세스 상태 확인하기. top보다 직관적.
$ nvidia-smi : GPU에서 실행중인 프로세스 상태 확인하기
$ cat (concatenate) : 지정한 파일의 내용 전체 출력
```

**\$ head** : 파일의 앞부분 출력, 기본 10줄 -n 옵션으로 첫번째 줄 부터 출력 할 범위 지정  
i.e) **head -n 20 file1.txt // file1.txt** 파일 첫번째 20줄 출력  
**\$ tail** : 파일의 뒷부분 출력, 기본 10줄, -n 옵션으로 마지막 줄 부터 출력 할 범위 지정  
i.e) **tail -n 20 file1.txt // file1.txt** 파일 마지막 20줄 출력  
**\$ more** : 창 하나에 보여지는 만큼만 출력  
Enter: 1줄 아래로, Space bar: 한 페이지 뒤로  
**\$ less** : 창 하나에 보여지는 만큼만 출력  
Enter: 1줄 아래로, Space bar: 한 페이지 뒤로, 방향키: 1행, 1열 이동  
page up/down: 페이지 이동, q: 끝내기

## 9. 문의 및 공용 프로그램 / 라이브러리 설치 신청

1) jhmoon@handong.ac.kr

본 사용자 지침서는 KISTI 국가슈퍼컴퓨팅 센터 누리온, 뉴론의 사용자 지침서, 전재영 교수님 빅모플 수업 영상을 참고하여 작성되었습니다.