

¿CÓMO TRABAJAR CON HPC?

HIGH-PERFORMANCE COMPUTING (COMPUTACIÓN DE ALTO RENDIMIENTO)

[Click acá para ir a documentación UAI](#)

1º Crear llaves de acceso

Según su dispositivo, copie y pegue el script entregado en la documentación --> Se crearán 2 llaves de manera automática. No las cambie de ubicación ni el nombre.

Generar llave pública y privada en Windows 10 vía cmd

En Windows 10 se encuentra el comando ssh-keygen instalado el cual generará los archivos id_ed25519.pub (llave pública) y id_ed25519 (llave privada) en el directorio del usuario desde una terminal cmd (tecla windows cmd)

Como activar ssh: https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui#install-openssh-for-windows

```
ssh-keygen -t ed25519 -C "hpc_uai"
```

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\demouai>ssh-keygen -t ed25519 -C "hpc_uai"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:/Users/demouai/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:/Users/demouai/.ssh/id_ed25519.
Your public key has been saved in C:/Users/demouai/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:LOmFFWUlkQXqQ5sN87FSUjFVm89wGFl8Tp6XnfhRNI hpc_uai
The key's randomart image is:
+--[ED25519 256]--+
|B+o*++|
|o B.o.o |
|= .E.o |
|o * ..-=|
|o So o ..|
|.. o ....o|
|. o ...+o |
| . o ..o |
|o .. . |
+---[SHA256]---
```

Ejemplo ssh-keygen en la terminal de windows

Generar Llave pública y privada en Linux/Mac

En Linux por defecto se encuentra el comando ssh-keygen el cual generará los archivos id_ed25519.pub (llave pública) y id_ed25519 (llave privada) en el directorio ~/.ssh del usuario

```
e https://linux.die.net/man/1/ssh-keygen
ssh-keygen -t ed25519 -C "hpc_uai"
```

```
demouai@zbook:~$ ssh-keygen -t ed25519 -C "hpc_uai"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/demouai/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/demouai/.ssh/id_ed25519.
Your public key has been saved in /home/demouai/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:5JyiyUVzohw0Kux+edYPjkmqWzz6FZJgdH+eTDpQ hpc_uai
The key's randomart image is:
+--[ED25519 256]--+
|o..|
|+= .o|
|+= o E .|
| == 0 B + .|
|..& X o S|
| o / + + .|
| . o + .|
| = .|
| o.o|
+---[SHA256]-----+
demouai@zbook:~$ ls -la .ssh/
total 16
drwx---- 2 demouai demouai 4096 may 28 14:47 .
drwxr-x-- 4 demouai demouai 4096 may 22 15:09 ..
-rw----- 1 demouai demouai 444 may 28 14:47 id_ed25519
-rw-r--r-- 1 demouai demouai 89 may 28 14:47 id_ed25519.pub
```

Para poder crear la cuenta: se debe enviar por correo a elias.estay@uai.cl la llave de carácter público (la que es id_ed25519.pub)

Tras creada la cuenta, Elías enviará 3 datos: ip (111.22.333.4), puerto (0123) y usuario (nnombre)

2º Ingresar al cluster

"Comando que permite conexión remota" se corre:

ej: ssh nnombre@111.22.333.4 -p 0123 --> ssh usuario@ip -p puerto

La primera vez se debe aceptar (yes) y pedirán crear clave (no la puedes ver así que cuidado). Si no deseas tener clave debes presionar ENTER sin escribir nada.

Linux/Mac

Al iniciar sesión por primera vez, se solicitará aceptar la llave del servidor una única vez (yes) e ingresar la clave de la llave privada.

```
ssh nombreUsuario@ip_publica -p puerto
```

```
demouai@zbook:~$ ssh demouai
The authenticity of host '111.22.333.4' is unknown.
ED25519 key fingerprint is SHA256:
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '111.22.333.4' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/demouai/.ssh/id_rsa':
Last login: Thu May 23 16:43:24 2024 from [REDACTED]
```

Windows

```
demouai@mansi:~$ ssh demouai
The authenticity of host '111.22.333.4' is unknown.
ED25519 key fingerprint is SHA256:
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '111.22.333.4' (ED25519) to the list of known hosts.
Enter passphrase for key 'C:/Users/demouai/.ssh/id_ed25519':
Last failed login: Tue May 28 16:06:24 -04 2024 from [REDACTED] on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Tue May 28 14:51:39 2024 from [REDACTED]
```

El ingreso normal (no nuevo) debería verse así (ej en Windows):

```
PS C:\Users\pulyp> ssh [REDACTED] ppizarro@[REDACTED]
[REDACTED]
Enter passphrase for key [REDACTED]
Last login: Thu Oct 17 12:49:27 2024 from 186.67.210.90
[REDACTED]

acceso primario:
ip: [REDACTED]
puerto: [REDACTED]

Cualquier duda o consulta la pueden enviar al correo elias.estay@uai.cl
Documentación oficial: https://wiki.mansci-web.uai.cl
[REDACTED]

[ppizarro@mansi ~]$
```

¡Felicitaciones! Accediste al HPC

Comandos básicos

pwd -> indica ubicación actual (al entrar a cluster debieses llegar a: /nfs_home/nombre).

ls -> listar objetos que hay en actual directorio.

cd -> para ir a un directorio específico. (ej: cd /nfs_home/nombre/carpeta, para ir a "carpeta")

exit -> desconectarse de cluster.

mkdir -> crear nuevo directorio (ej: mkdir folder_new, crea una carpeta "folder_new")

rm -> elimina archivo (ej: rm script.py)

(rm -r: directorio / rm *.out: todo lo que termina en .out / etc)

nano -> editor de texto para modificar archivo desde terminal

pip3 install -> instalar librerías

pip3 list -> listar librerías instaladas

SINFO

```
[ppizarro@mansi ~]$ sinfo
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute-fat    up    7-00:00:00      2  idle  compute-fat-0-[16-17]
compute-slim*   up    7-00:00:00      6  alloc  compute-slim-0-[0,2,5,8,10,12]
compute-slim*   up    7-00:00:00      6  idle  compute-slim-0-[1,3,7,9,11,13]
compute-xs      up    7-00:00:00      4  alloc  compute-xs-1-[0,2,4,6]
compute-xs      up    7-00:00:00      3  idle  compute-xs-1-[1,3,5]
compute-fit     up    7-00:00:00      1  mix   compute-fit-2-5
compute-fit     up    7-00:00:00      2  alloc  compute-fit-2-[6-7]
compute-fit     up    7-00:00:00      3  idle  compute-fit-2-[2-4]
compute-gpu     up    7-00:00:00      2  mix   compute-gpu-2-[8-9]
[ppizarro@mansi ~]$
```

sinfo -> indica info sobre particiones y estado de nodos

squeue -> muestra TODOS los trabajos pendientes/corriendo (squeue -u nombre: ver los jobs propios)

sbatch -> envía script de trabajo .slurm a la cola (ej: sbatch myjob.slurm)

scancel -> cancela trabajos (scancel ID, scancel -u nombre)

sacct -> Muestra detalles sobre trabajos completados

(ej para solicitar formato específico: sacct --format=JobID,JobName,Partition,Elapsed,MaxRSS,State)

ID Cuando se envía un trabajo a la cola, se crea un JobID para ese trabajo

```
[ppizarro@mansi ~]$ sacct --format=JobID,JobName,Partition,Elapsed,MaxRSS,State
  JobID  JobName  Partition  Elapsed  MaxRSS  State
-----
```

scp -> copia archivos entre dispositivo local y clúster (SE CORRE EN COMANDO DE COMPUTACIÓN PERSONAL)

(Ej para copiar de computadora a HPC: scp archivo.txt usuario@ip:/ruta/destino/)

(Ej para copiar de HPC a computadora: scp usuario@ip:/ruta/origen/archivo.txt /ruta/local/)

scp -r (copia carpeta)

¿Cómo enviar trabajos a la cola? (Lanzar trabajos para que corran)

```
myjob.slurm
#!/bin/bash
## save this file as myjob.slurm
## nodes request and configuration
#SBATCH --job-name=mydemojob
#SBATCH --ntasks=4
#SBATCH --time=01:30:00
#SBATCH --partition=compute-slim
#SBATCH --nodelist=compute-slim-0-0
## job Name
## Total number of tasks requested
## Run time (hh:mm:ss) - 1.5 hours
## Name of partition
## Run in specific node

## output
#SBATCH --output=%u_%x_%j.out
## output userName_jobName_jobId.out

# Put your software in this line
hostname
```

Ejecutaremos desde nuestra cuenta el comando " sbatch" y el archivo "myjob.slurm" lo que nos entregará el id asignado a nuestro job

```
[demouai@mansi ~]$ sbatch myjob.slurm
Submitted batch job 24604
```

- Crea un script de trabajo (.slurm)
-> Este script es un archivo donde indicas al sistema qué recursos necesitas y qué comandos deseas ejecutar.
- Para especificar recursos se parte la línea del código con: #SBATCH
- Un ejemplo es el de la imagen de la izquierda.
- Este script lo guardas como: myjob.slurm
- Para enviarlo: sbatch myjob.slurm

- Ejemplo de tarea para corrección de sesgo de data climática ----->
- Es muy importante comprender todas las indicaciones que se irán a dar. La cantidad de memoria y nodos, etc se van conociendo a prueba y error
- --output o --error: es para solicitar el lugar de guardado de los archivos que indican los prints o errores de tu job.

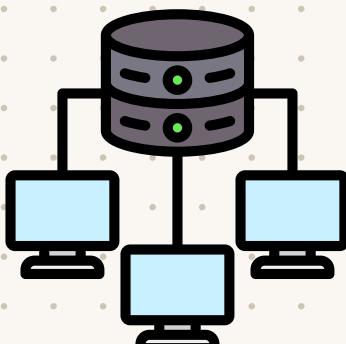
```
#!/bin/sh
## save this file as myjob.slurm
## nodes request and configuration

#SBATCH --partition=compute-fit
#SBATCH --job-name= Divide_Zone_Bias
#SBATCH --ntasks-per-node=1
#SBATCH --nodes=2
#SBATCH --cpus-per-task=128
#SBATCH --mem=512G
#SBATCH --time=2-00:00:00
## Nombre de la particion a usar
## Nombre del trabajo
## 1 proceso x nodo
## 2 nodos a utilizar
## 128 hilos x proceso
## Mmeoria x nodo
## Tiempo maximo de ejecucion (2 dias)

#SBATCH --output=/nfs_home/ppizarro/carpeta_de_outputs/%j.out
#SBATCH --error=/nfs_home/ppizarro/carpeta_de_outputs/%j.err
```

IMPORTANTE: en tu script debes tener una función/indicación para crear un archivo con tus resultados.

IMPORTANTE: crear un environment para llamar desde el script .slurm (ya sea python, anaconda, etc).



Ejemplo de complicaciones:
(Corrección de sesgo para data climática modelada)

- ¿Script no encuentra librerías como numpy? A pesar de estar todas instaladas tanto en el HPC como en tu propio environment.
- ¿Tampoco encuentra tu environment?
- Solución en caso de utilizar un nro pequeño de librerías:
Desde tu script .slurm llamar a crear el environment y llamar a instalar las librerías/cargar módulos necesarios.
- echo: equivale a un print que podrás ver en tus outputs para así saber que el script avanzó sin problemas (en este caso).

- Lo más probable es que existan mejores soluciones, personalmente esta fue la forma más simple que encontré de arreglar estas complicaciones para este script.
- Siempre pueden solicitar ayuda vía mail a: ayudahpc@uai.cl

```

echo "Activating python module"
ml rhpython/3.8

echo "Currently using this python $(which python)"

echo "Creating python environment"
python -m venv my_env
source ./my_env/bin/activate

echo "Installing python packages"
python -m pip install --upgrade pip
python -m pip install numpy
python -m pip install scipy
python -m pip install matplotlib
python -m pip install pandas
python -m pip install seaborn
python -m pip install xarray
python -m pip install netCDF4
python -m pip install dask
python -m pip install dask-jobqueue
echo "Installed python packages"

#cargar modulos necesarios
module load dask-jobqueue

# Ejecutar tu script de Python
echo "STARTING SCRIPT"
srun python ./DivideZonesBiasMejorGuardado.py
|
```

```

[ppizarro@mansi ~]$ ls
2_0_BiasCorrectionSST.py  BiasCorrectionSST.py  BiasMiniZone_Dask.py
climQMBC2                 DivideZonesBias.py  myminijob.slurm
Probando.py                bias_environment      carpeta_de_outputs
                           my_env                   myprueba.slurm
3Points_Bias.py            BiasFutHis.py       climQMBC
Data_sst_HPC               DivideZonesBiasMejorGuardado.py  perl5
                           Bias_5points.py    myjob.slurm
                           DivideZonesBiasMejorGuardado.py
                           [ppizarro@mansi ~]
                           [ppizarro@mansi ~]
                           [ppizarro@mansi ~]$ ls climQMBC
                           __init__.py  __init__.pyc  methods.py  methods.pyc  __pycache__  repo
                           rt.py        utils.py    utils.pyc
                           [ppizarro@mansi ~]$ |
```

- Para este script de corrección de sesgo se trabajó con la librería climQMBC, la cual se duplicaba desde GitHub (con gitclone).
- El script no lograba importar las funciones que esta librería posee, no las encontraba.
- Solución: Se duplicó la carpeta climQMBC -> con la data respaldada se procedió a eliminar elementos de climQMBC, dejando únicamente los archivos .py a la mano, así el script llegaba directamente a las funciones .py que desea importar.