



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



EXCELENCIA
SEVERO
OCHOA

How to use Marenostrum 5

Jon Navarro
Maria Velez de Villa
Felix Fernando Ramos

Barcelona, May 23rd 2024

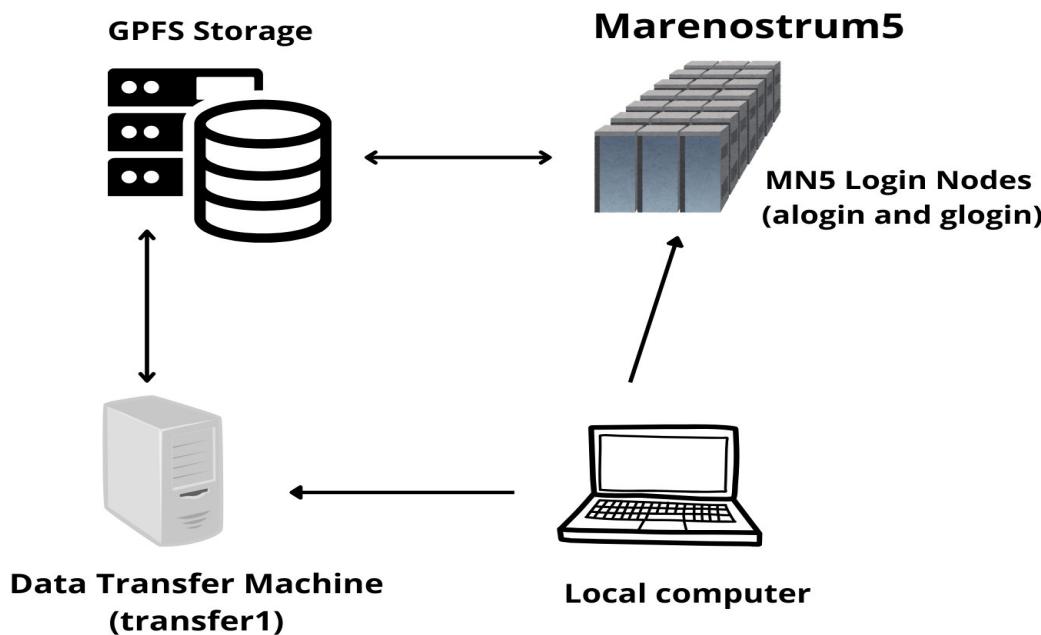
Basic Usage Guide

Overview

- **Access to Marenostrum**
- **Marenostrum Logins**
- **Filesystems**
- **Data Transfer**
- **Compilers**
- **Batch system**

Shared Credentials

- Shared users among:
 - Data Transfer
 - MareNostrum 5 (GPP and ACC)
 - Some other HPC machines
- Shared filesystem (GPFS)
- Same username (\$USER) & passwd (centrally managed)

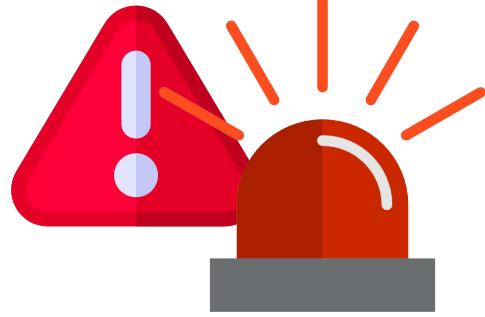


BSC HPC Access

- **Access through SSH**
 - OpenSSH for Linux / macOS
 - PuTTY for Windows
- **On the facilities granted you can authenticate by:**
 - password
 - SSH public keys
- **How to generate & use keys:**
 - \$ ssh-keygen -t rsa
 - \$ ssh-copy-id **nct01XXX@glogin1.bsc.es**



MareNostrum logins

<ul style="list-style-type: none">● 4 external accessible logins:<ul style="list-style-type: none">○ glogin1.bsc.es○ glogin2.bsc.es○ alogin1.bsc.es○ alogin2.bsc.es	<ul style="list-style-type: none">● 2 internal accessible logins:<ul style="list-style-type: none">○ glogin4.bsc.es○ alogin4.bsc.es
<ul style="list-style-type: none">● No outgoing connections No downloads or uploads● 5 minutes cpu time limit Use transfer1 node for long data transfers. For long compilations, use interactive sessions: \$ salloc -A [account] -q [quos] -p interactive	

Login node usage

		
Manage and edit files		Run production executions
Small and medium compilations		Copy large amount of files
Submit jobs to batch system		Long and heavy load graphical interfaces
Check results and prepare scripts		Launch cheeky fork bombs

General Parallel Filesystem (GPFS)

Relevant filesystems to your everyday use:

- **/apps/GPPI** and **/apps/ACCI** (Applications and libraries)
 - Applications installed on MN.
- **/gpfs/home** (User's home)
 - Scripts, codes and documents.
- **/gpfs/projects** (Data, custom installations)
 - Execution folder, init data, shared data in your group.
- **/gpfs/scratch** (Temporary files, **doesn't have backups**)
 - Execution data, huge log/output files, temporary files.
- **/gpfs/tapes/hpc** (Archive): HSM (Hierarchical Storage Management)

/gpfs/home Filesystem usage

- Few space (~80 GB per user)

- **/gpfs/home Do:**

- Store source code
 - Store personal scripts



- **/gpfs/home Don't:**

- Use as production directory



/gpfs/projects Filesystem usage

- You can check availability with `bsc_quota`

- **/gpfs/projects Do:**

- Use as production directory
 - Shared data in your group



- **/gpfs/projects Don't:**

- Store source code
 - Store personal scripts



/gpfs/scratch Filesystem usage

- You can check availability with `bsc_quota`

- **/gpfs/scratch Do:**

- Use as production directory
- Temporary files



- **/gpfs/scratch Don't:**

- Store source code
- Store personal scripts



Doesn't have backups

General Parallel Filesystem (GPFS)

IMPORTANT:

It is your responsibility as a user of our facilities to backup all your critical data. We only guarantee a daily backup of user data under /gpfs/home and /gpfs/projects . Any other backup should only be done exceptionally under demand of the interested user.

Filesystem limits (Quota)

- Filesystem limit per user and/or group.
- Check with: `bsc_quota`

bsc_quota							files	in_doubt
Filesystem	type	blocks	quota	limit	in_doubt	grace		
gpfs_home	USR	52.95G	78.15G	79.15G	20.98M	none	761795	439
gpfs_projects	GRP	23.30T	24.41T	25.63T	41.94G	none	29378841	102289
gpfs_scratch	GRP	15.09T	19.53T	20.51T	79.22G	none	10341779	20079

- Typical related problems:
 - Job submission failure when `$HOME` is over quota
 - Job execution failure when writing to over-quota filesystem
- Group-SHARED quota on `/gpfs/projects` and `/gpfs/scratch`

GPFS Filesystem Performance

Useful tips to achieve optimal I/O performance:

- **Stay below your assigned quota**
 - There is performance degradation when you get near the quota limit
- **Keep a low file number per directory**
 - Keep below 1000 files per directory if possible
- **Avoid creating/writing many files at the same time and place**
 - Will compromise I/O speed
- **Avoid small files (block size of 16 MB in projects and scratch)**
 - These block sizes are divided into 1024 subblocks, meaning that the smaller file will weight at least 16 KB

Node's local disk (/scratch)

- **Each node has a local disk for temporary files**
 - Accessible via `$TMPDIR` (this env variable points to the temporary directory)
 - Not shared between nodes (different to `/gpfs/scratch`)
 - Content erased after related job execution
- **Useful for temporary files**
 - Temporal data used by a job assigned to that node
- **876 GB, SSD (GPP)**
- **436 GB, SSD (ACC)**

How to transfer data to MN5

- MN5 logins have 5 minutes of CPU time limit to avoid executions
- MN5 does not allow outgoing connections
- You must use the data transfer node to transfer files (transfer1.bsc.es). Widely used commands (from your local machine) are:
 - **scp**
 - scp localfile username@transfer1.bsc.es:/path/remote/dir
 - scp username@transfer1.bsc.es:/path/remote/file localdir
 - **rsync**
 - rsync -av /path/localdir
username@transfer1.bsc.es:/path/remotedir
 - **sshfs, sftp, bbcp, ftps**

Data Transfer Commands

- Set of commands to send data transfer jobs to queues
- Available in transfer1 and MareNostrum (on MN5, load module transfer)
- Available commands:

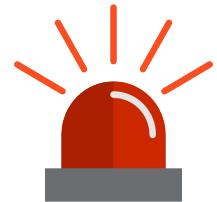
File movement	Archiving and synchronizing	Job Control
dtcp Copying files	dttar Compressing files	dtq Check datajobs status
dtrmv Moving files	dtrsync Synchronizing files	dtcancel Cancel datajobs

- More dt commands and information via **man dtcommands**

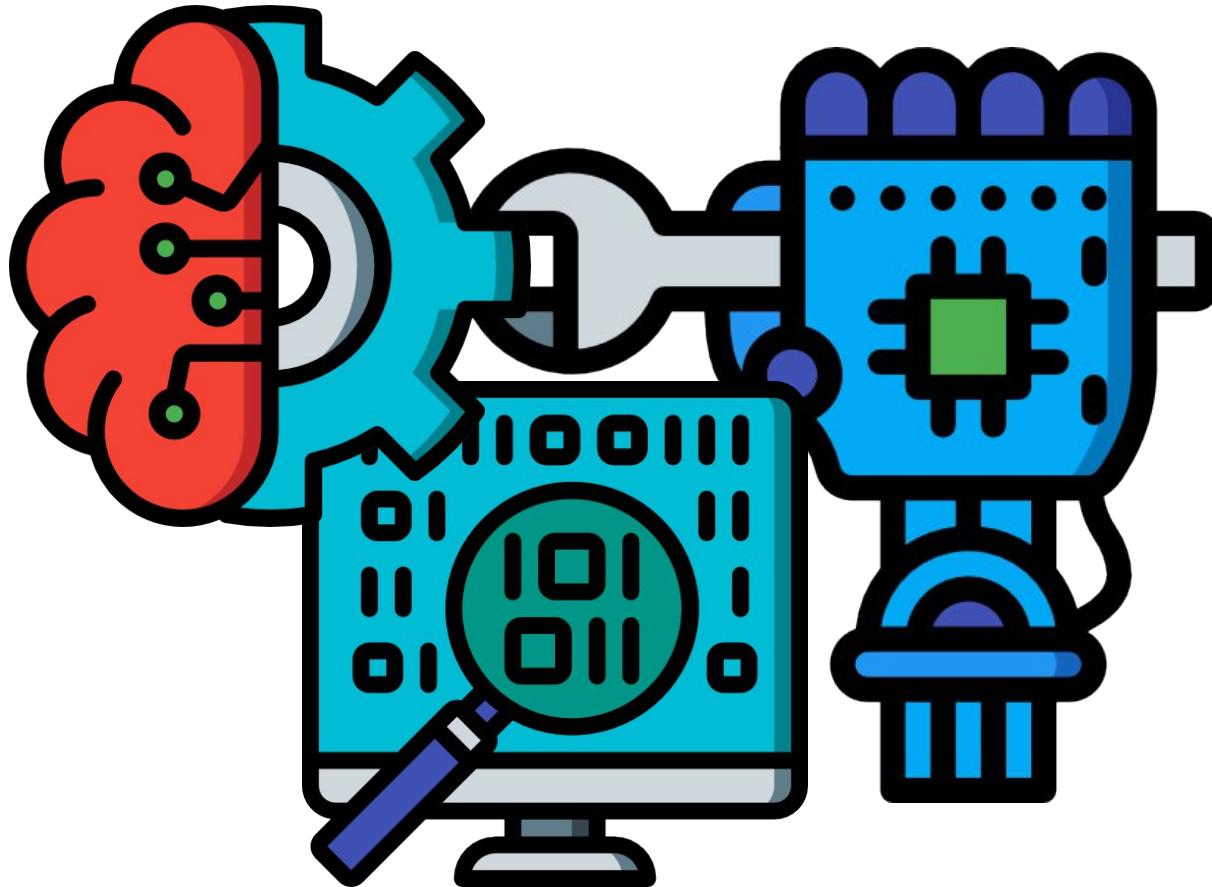
Managing permissions for data access

- **Do not alter basic access permissions**
 - Unwanted access, accidental alteration/deletion
 - Coarse granularity
- **Use ACLs**
 - Per user/group access to files and directories
 - Small granularity
- **When applying changes to access permissions be aware of the group (primary vs account group) from where you are working.**
- **When in doubt, ask Support Team**

Check: -> <https://www.bsc.es/supportkc/FAQ>



Hands-on 1.1 and 1.2: Changing the password and transferring files



Module Environment (I)

- System used by Marenostrum to manage all installed software
- Environment variables and software dependencies management
- Several versions of the same program coexisting at `/apps/GPP` and `/apps/ACC/`
- If you need additional software, you can ask support to install it as a module
- Default modules loaded will depend on the partition:
 - Intel compiler, libraries and tools (`intel/2023.2.0`) - GPP
 - Intel MKL (`mkl/2023.2.0`) and Intel MPI (`impi/2021.10.0`) - GPP
 - BSC custom commands (`bsc/1.0`) - GPP and ACC

Module Environment (II)

- **Module commands:**

Command	Option	Example	Info
avail	[program]	module avail	List available modules
list	[program]	module list	List loaded modules
purge		module purge	Unload all modules
load/unload	<program[/version]>	module load/unload gcc/5.1.0	Load or unload a module
switch	<old> <new>	module switch intel gcc	Change a module by another

Compilers

- Intel, GNU and NVIDIA compiler suites available via modules
- Several versions, managed by the module system
 - Intel (licensed)
2023.0, 2023.1, 2023.2.0, 2024.0
 - GCC (Free Software)
11.4.0, 13.2.0
 - NVIDIA (licensed) - Only for ACC
23.9, 23.11, 24.3
- MPI compilation also managed by modules through wrappers
 - Intel: mpicc (C), mpiicpc (C++), mpifort (FORTRAN), ...
 - GCC: mpicc (C), mpicxx (C++), mpifort (FORTRAN), ...
 - NVHPCX: mpicc (C), mpicxx (C++), mpifort (FORTRAN), ...
- Load optimization flags for compiling:

Compiler optimization drawbacks

- **Each software is different, but:**
 - Intel can get up to 20% performance increase in some applications
 - Linking mkl libraries usually boosts performance
 - Static compilation sometimes runs faster
- **Optimization drawbacks**
 - Over optimization may result in numeric error
- **Intel compilers might get a bit finicky with some compilations, you could try to use GCC instead and vice versa**

Batch System

- MareNostrum 5 uses SLURM as batch system
- Benefits of using jobscripts
 - Defines resources needed
 - Reusable
 - Document needs and requests
 - Jobscripts are shell scripts with special markings
- Each submission is a job

```
> sbatch test_job  
Submitted batch job 8936247  
> squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
8936247	main	test_job	nct01026	R	0:19	1	s02r1b36



SLURM commands

- **Submit a job defined in a file**

- `sbatch -A [accountName] -q [qosName] jobsript`

```
> sbatch test_job
```

```
Submitted batch job 8936247
```

- **Check status of jobs submitted:**

- `squeue`

```
>squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
8936247	main	test_job	nct01026	R	0:19	1	s02r1b36

- **Cancel a job:**

- `scancel JobID`

```
> scancel 8936247
```

Job queues

- **Jobs are assigned to queues (QoS)**
 - You have to select the queue to run the job
- **Specify different kind of queues: gp_debug, gp_interactive, gp_graphical. Each queue has different purposes and limitations.**
- **There are different queues per partition: gp_debug, acc_debug**
- **Check your available queues and their limits:**
 - **bsc_queues**

SLURM Common Parameters

Option	Comment	Example
-n --ntasks	Number of tasks	#SBATCH -n 32
-t --time	Wallclock limit	#SBATCH -t 01:00
-J --job-name	Job name	#SBATCH -J myjob
-A --account	Account name	#SBATCH -A bsc99
--gres=gpu:N	Gpus per node	#SBATCH --gres=gpu:4
-o --output	Output file	#SBATCH -o %j.out
-e --error	Error file	#SBATCH -e %j.err
-q --qos	Queue	#SBATCH -q=gp_debug
--exclusive	Exclusive mode	#SBATCH --exclusive
-D --workdir	Current working dir	#SBATCH -D /my/path/

SLURM Extra Parameters: Process layout

- How to define specific load balance configurations:

Option	Comment	Example
--ntasks-per-node	Tasks per node	#SBATCH --ntasks-per-node=48
-c --cpus-per-task	Cpus per task	#SBATCH -c 1

SLURM Extra Parameters: Memory Requirements

Two types of memory configuration:

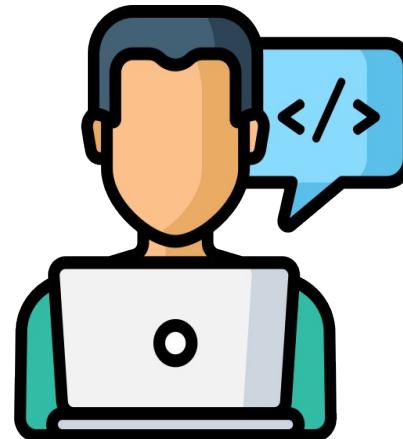
- **STANDARD NODES**
 - Total of 256 GBytes per node in GPP and 512 GBytes in ACC
- **HIGH MEMORY NODES (GPP)**
 - Total of 1024 GBytes per node
 - Only 216 nodes available
 - `#SBATCH --constraint=highmem`
 - Requesting them will make you wait more in the queue
- The default behaviour is a lowmem node.

Job Examples: Sequential

Sequential example

```
#!/bin/bash
#SBATCH --job-name=seq_job
#SBATCH --chdir=.
#SBATCH --output=serial_%j.out
#SBATCH --error=serial_%j.err
#SBATCH --ntasks=1
#SBATCH --time=00:02:00

./serial_binary
```



Job Examples: Threaded

Threaded example (OpenMP, pthreads, ...)

```
#!/bin/bash
```

```
#SBATCH --job-name=omp_job
#SBATCH --chdir=.
#SBATCH --output=omp_%j.out
#SBATCH --error=omp_%j.err
#SBATCH --cpus-per-task=112
#SBATCH --ntasks=1
#SBATCH --time=00:10:00
#SBATCH --qos=gp_debug
```

```
export SRUN_CPUS_PER_TASK=${SLURM_CPUS_PER_TASK}
```

```
./openmp_binary
```

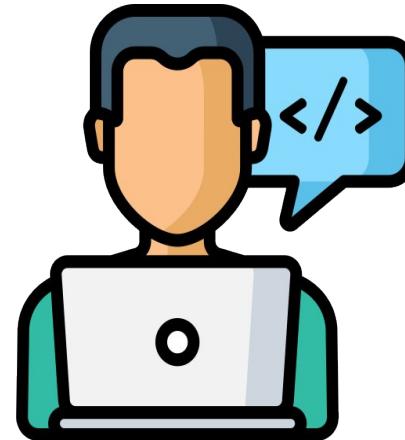


Job Examples: typical pure MPI

MPI example (multiple nodes, Intel MPI)

```
#!/bin/bash
#SBATCH --job-name=mpi_job
#SBATCH --output=mpi_%j.out
#SBATCH --error=mpi_%j.err
#SBATCH --ntasks=224

srun ./mpi_binary
```



Job Examples: hybrid MPI + OpenMP

MPI + Threads example

```
#!/bin/bash
#SBATCH --job-name=hybrid_job
#SBATCH --chdir=.
#SBATCH --output=mpi_%j.out
#SBATCH --error=mpi_%j.err
#SBATCH --ntasks=56
#SBATCH --cpus-per-task=4
#SBATCH --tasks-per-node=28
#SBATCH --time=00:02:00
```

```
export SRUN_CPUS_PER_TASK=${SLURM_CPUS_PER_TASK}
```

```
srun ./hybrid_binary
```



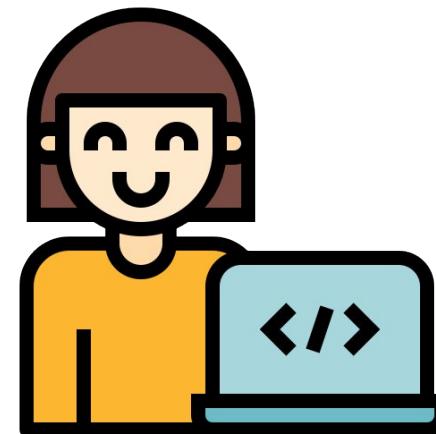
Job Examples: GPUs in ACC

GPUs example

```
#!/bin/bash
#SBATCH --job-name=gpu_job
#SBATCH -D .
#SBATCH --output=mpi_%j.out
#SBATCH --error=mpi_%j.err
#SBATCH --ntasks=80
#SBATCH --cpus-per-task=2
#SBATCH --time=00:02:00
#SBATCH --gres=gpu:4
```

```
export SRUN_CPUS_PER_TASK=${SLURM_CPUS_PER_TASK}
```

```
srun ./gpu_binary
```



SLURM Jobs Comparison - GPP

	CPUs/task	task/node	tasks
Sequential	1	1	1
OpenMP	112	1	1
MPI	1	112	112 * num nodes
MPI + OpenMP	112	1	num nodes

SLURM Jobs Comparison - ACC

	CPUs/task	task/node	tasks
Sequential	1	1	1
OpenMP	160	1	1
MPI	1	160	160* num nodes
MPI + OpenMP	160	1	num nodes

Best practices when contacting Support

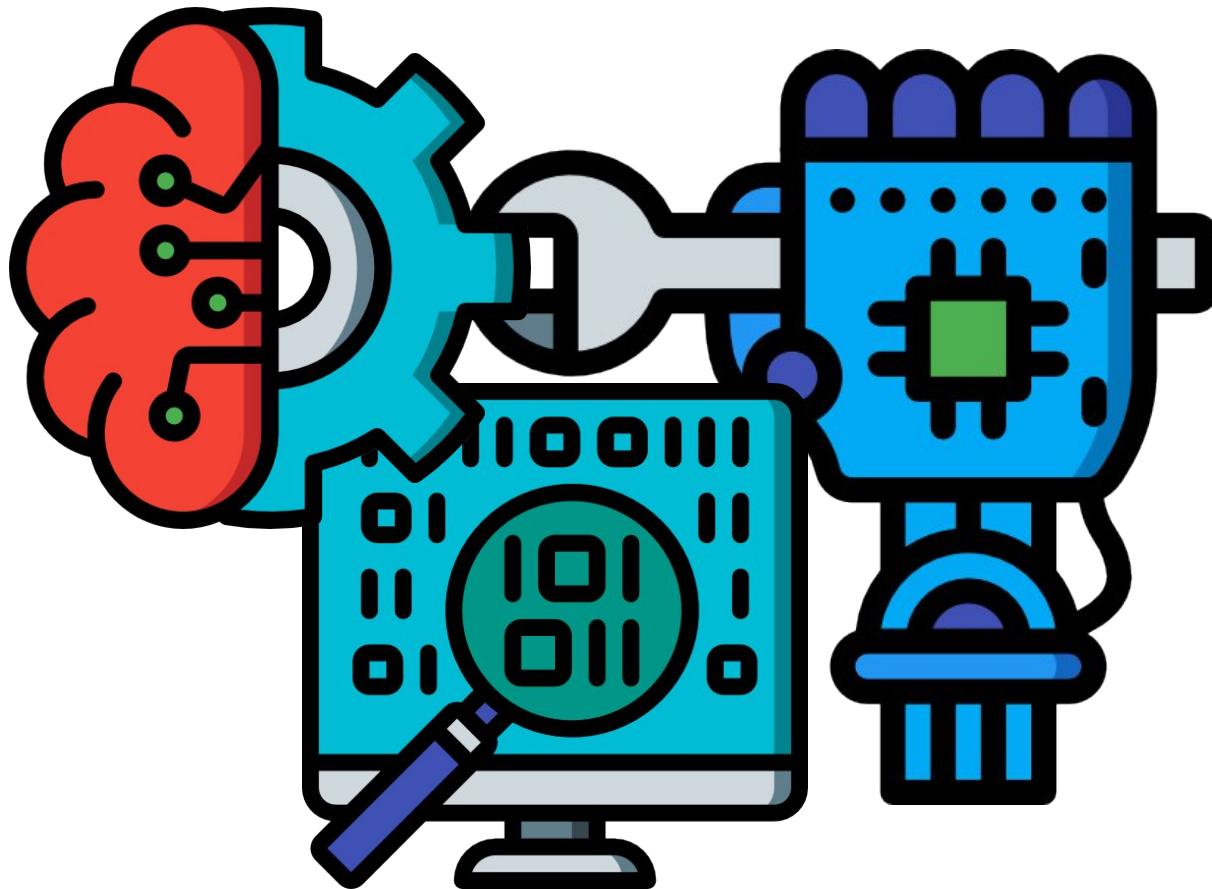
- When contacting support remember this:

- Specify
 - Partition
 - Job Ids
 - Software version
 - Environment (if applies)
 - Machine
 - Username
 - Exact steps that lead to the issue
 - Error messages
- Do not take for granted that we know what you know, want or need. Effective communication results in a faster resolving time.
- We don't know who you are but we will try our best to help you, we have no favorites



*Average supporter after receiving a ticket
without any details whatsoever
BSC 2021, recolorized*

Hands-on 1.3, 1.4 and 1.5: Modules, SLURM resources and Compilers





**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



EXCELENCIA
SEVERO
OCHOA

Thank you

support@bsc.es