

Instrucciones de clúster de computación IZARO

El clúster de computación IZARO está administrado en conjunto por la Plataforma de Bioinformática, Bioestadística y Sistemas de Información y con el soporte técnico de HPCnow!. Para dudas respecto al uso del clúster, problemas técnicos, instalación de módulos, entre otros, contactad con la plataforma a través de los siguientes correos:

informatica@bio-bizkaia.eus

bioinformatica@bio-bizkaia.eus

1.	Características del clúster	1
2.	Alta de grupo y usuario	2
2.1	Tarifas	2
3.	Conexión y primeros pasos	2
4.	Sesión interactiva	3
5.	Envío de trabajos a nodos de cómputo	3
5.1	Ejecutar trabajo en tiempo real (con srun)	3
5.2	Trabajos batch con scripts (usando sbatch)	3
5.3	Comandos útiles para el seguimiento y manejo de trabajos	4
6.	Uso de módulos (ejemplo con Miniforge3)	4
7.	Uso de contenidos (Apptainer)	6

1. Características del clúster

El clúster de computación consta de las siguientes características:

- 1 Nodo de administración X430 A6. 1 procesador AMD Turin 9655 con 96 cores a 2,6GHz de frecuencia. 128 GB de RAM y 2 discos SSD 0.96 TB.
- 3 Nodos de cómputo Thin Bull Sequana X440 A6. Cada uno con 2 procesadores AMD Turin 9655 con 96 cores a 2,6GHz de frecuencia. 1152 GB de RAM (6 GB por core) y 2 discos SSD 0.96 TB.
- 2 Nodos de cómputo Fat Bull Sequana X430 A6. Cada uno con 1 procesador AMD Turin 9655 con 96 cores a 2,6GHz de frecuencia. 1152 GB de RAM (12 GB por core) y 2 discos SSD 0.96 TB.
- 1 Nodo de cómputo GPU Bull Sequana X450 A6. 1 Nvidia L40S. 1 procesador AMD Turin 9655 con 96 cores a 2,6GHz de frecuencia. 1152 GB de RAM (6 GB por core) y 2 discos SSD 0.96 TB.
- Sistema de almacenamiento de alto rendimiento con 830 TB netos en scratch tipo Beegfs. Los servidores de almacenamiento son 2 Bull Sequana X430 A6, cada uno con 1 procesador AMD Turin 9655 con 96 cores a 2,6GHz de frecuencia. 128 GB de RAM y 2 discos SSD 0.96 TB.
- Sistema de almacenamiento tipo cold de 220 TB.

2. Alta de grupo y usuario

Las cuentas del clúster se estructuran mediante grupos. Un investigador principal o un responsable de plataforma adscrito al IIS Biobizkaia puede solicitar el alta de su grupo mediante la web de Biobizkaia:

<https://www.bio-bizkaia.eus/plataformas/hub-tecnologico/bio2si#formulario>

Una vez realizada la solicitud, se creará el home privado de cada usuario:

/shared/home/<nombre_usuario>

Además, se creará un grupo de Linux y un directorio común para el grupo:

/shared/scratch/<nombre_grupo>

Dado que ambos directorios se encuentran en el sistema de almacenamiento de alto rendimiento, a nivel de rendimiento es indiferente el usar el Home o el almacenamiento común de grupo.

Atención: En el directorio común de grupo, todos los ficheros y directorios creados serán accesibles por los miembros del grupo, por lo que se recomienda **revisar antes de eliminar o modificar** archivos en ese directorio. Para simulaciones individuales sin necesidad de ser compartidas, se recomienda usar el directorio home.

La **cuota de almacenamiento** correspondiente es común al grupo, y vendrá determinada por defecto a **2 TB** por cada usuario hasta un **máximo de 4**, dando un **máximo total de 8 TB**. Pudiendo solicitar más almacenamiento extra acorde a las tarifas. Por su parte, el número de cuentas asociadas podrá ser superior a 4 en función de las necesidades del grupo sin coste extra.

Este almacenamiento será compartido por el grupo y será el propio grupo el responsable de administrarlo, sin estar limitado por usuario individual, e independientemente de si se ocupa en el Home privado o el almacenamiento común de grupo.

En caso de necesitar un espacio de almacenamiento común entre distintos grupos, consultad con la plataforma.

2.1 Tarifas

Las tarifas presentadas a continuación son provisionales y se comenzarán a aplicar a partir del 1 de marzo de 2026.

Concepto	Importe
Almacenamiento	0.01 €/GB al mes
Cómputo	0.005 €/core*hora

3. Conexión y primeros pasos

El acceso al HPC se hace únicamente desde la red **Biocruces-Inves**. En caso de no estar en dicha red, será necesaria una VPN para el acceso, gestionado por la Unidad de Sistemas de Información de Biobizkaia.

Una vez creado el usuario, accederemos al nodo de Login mediante ssh en un terminal en caso de disponer de Linux o Mac, o bien mediante software como Bitwise o Putty.

Ejemplo de ssh en terminal:

```
[usuario@local ~]$ ssh usuario@192.168.8.15
```

Nos solicitará nuestra contraseña recibida por correo, que deberá ser cambiada ejecutando el siguiente comando:

```
[usuario@login01 ~]$ passwd
```

4. Sesión interactiva

El nodo de login no está indicado para la realización de trabajos, por lo que podemos iniciar una sesión interactiva en uno de los nodos de cómputo con 2 cores y 4 GB de RAM por defecto con el comando "salloc". Es recomendable elegir los cores y la RAM deseadas, así como el tipo de nodo y el tiempo que se va a reservar la sesión interactiva. Los tipos de nodos y sus características se muestran en la siguiente tabla:

Nodo	Tipo	CPUs totales	Memoria total
c01-c03	THIN	96 x 2	~1.16 TB
f01-f02	FAT	96 x 2	~1.16 TB
g01	GPU	96 x 2	~1.16 TB

Ejemplo: Para solicitar una sesión en el thin node c01 con 8 GB de RAM y 2 cores durante 1h:

```
[usuario@login01 ~]$ salloc -p main --nodelist=c01 -c 2 --mem=8G -t 01:00:00
salloc: Pending job allocation 168
salloc: job 168 queued and waiting for resources
salloc: job 168 has been allocated resources
salloc: Granted job allocation 168
salloc: Nodes c01 are ready for job
```

Para salir y liberar recursos:

```
[usuario@c01 ~]$ exit
```

5. Envío de trabajos a nodos de cómputo

En SLURM, los métodos más habituales para ejecutar trabajos son srun y sbatch.

5.1 Ejecutar trabajo en tiempo real (con srun)

Ejecuta un programa directamente asignando recursos, sin scripts. Por ejemplo, para ejecutar un programa llamado mi_programa con 8 cores y 16 GB RAM durante 2 horas:

```
[usuario@login01 ~]$ srun -p main -c 8 --mem=16G -t 02:00:00 ./mi_programa
```

5.2 Trabajos batch con scripts (usando sbatch)

Lo habitual para trabajos más grandes o repetitivos es crear un script que contenga todas las indicaciones de recursos y comandos. SLURM gestionará cuándo y dónde ejecutarlo.

Ejemplo de script mi_trabajo.slurm:

```
#!/bin/bash
#SBATCH -J mi_trabajo # Nombre del trabajo
```

```
#SBATCH -p main # Partición o cola
#SBATCH -N 1 # Número de nodos (1 nodo)
#SBATCH -c 8 # Número de cores
#SBATCH --mem=32G # Memoria RAM total
#SBATCH -t 04:00:00 # Tiempo máximo (4 horas)
#SBATCH -o salida_%j.out # Archivo para salida estándar
#SBATCH -e error_%j.err # Archivo para salida de errores

module load Python/3.13.1-GCCcore-14.2.0 # Carga módulos necesarios
```

Para enviar el trabajo:

```
[usuario@login01 ~]$ sbatch mi_trabajo.slurm
Submitted batch job 456789
```

SLURM asignará recursos cuando estén disponibles y ejecutará el script. La salida y errores irán a los archivos indicados (salida_456789.out y error_456789.err).

5.3 Comandos útiles para el seguimiento y manejo de trabajos

Ver trabajos en cola o ejecución:

```
squeue
```

Cancelar trabajo:

```
scancel <job_id>
```

Ver detalles de un trabajo:

```
scontrol show job <job_id>
```

Estado de las particiones y nodos:

```
sinfo
```

6. Uso de módulos (ejemplo con Miniforge3)

Miniforge3 está disponible como módulo en los nodos de cómputo del clúster. Por tanto, para cargarlo y usarlo es necesario previamente abrir una **sesión interactiva** (ver sección 5 de este manual) en un nodo de cómputo, ya que en el nodo de login no está disponible. Una vez estamos en un nodo de cómputo:

Cargar el módulo Miniforge3

```
[usuario@c02 ~]$ module load Miniforge3
```

Crear un entorno con Python 3.11:

```
[usuario@c02 ~]$ conda create -n mi_entorno python=3.11
```

La primera vez que intentes activar un entorno con conda activate <env>, es posible que recibas un mensaje solicitando ejecutar:

```
conda init
```

Esto configura Conda para funcionar correctamente en el shell. Tras ejecutarlo, deberás cerrar y volver a abrir la sesión interactiva para que los cambios surtan efecto.

```
[usuario@c02 ~]$ conda init
```

```
...
```

```
[usuario@c02 ~]$ exit
```

Luego hay que volver a abrir una interactiva mediante salloc y cargar el módulo.

Activar el entorno:

```
[usuario@c02 ~]$ conda activate mi_entorno
```

```
(mi_entorno) [usuario@c02 ~]$
```

Para salir del entorno:

```
(mi_entorno) [usuario@c02 ~]$ conda deactivate
```

```
[usuario@c02 ~]$
```

Instalar paquetes dentro del entorno activado:

```
(mi_entorno) [usuario@c02 ~]$ conda install numpy scipy pandas
```

Usar Miniforge3 en scripts SLURM:

```
#!/bin/bash
#SBATCH -J trabajo_conda
#SBATCH -p main
#SBATCH -c 8
#SBATCH --mem=32G
#SBATCH -t 04:00:00
#SBATCH -o salida_%j.txt

module load Miniforge3

eval "$(conda shell.bash hook)" # Necesario para evitar que nos pida
conda init

conda activate mi_entorno

python mi_script.py
```

7. Uso de contenidos (Apptainer)

Apptainer es la herramienta instalada por defecto en el clúster para ejecutar contenedores, por lo que no es necesario cargarla como módulo.

Como en el caso de Miniforge3, la utilizaremos desde una sesión interactiva en un nodo de cómputo (sección 5 de este manual).

Ejemplo de ejecución de fastqc desde apptainer:

```
[usuario@c02 ~]$ apptainer pull fastqc_v0.11.9_cv8.sif docker://biocontainers/fastqc:v0.11.9_cv8
...
[usuario@c02 ~]$ apptainer exec fastqc_v0.11.9_cv8.sif fastqc --version
FastQC v0.11.9
```

Versión	Realizado	Revisado	Aprobado
01 – 14/10/2025	Bioinformática, bioestadística y sistemas informáticos	Bioinformática, bioestadística y sistemas informáticos	Calidad