

Estados de los job en SLURM

sbatch -p special -N2 -n2 ejemplo.sh → lanzar trabajos
srun -p generic -N1 -n1 --pty \$shell -i → lanzar en interactivo
srun -p gpu -N1 -n1 --pty \$shell -i → lanzar en interactivo en gpu
scancel <jobid> → cancela el trabajo
squeue → ver mi cola de trabajos
squeue -p "partición" "nodo" → Ver trabajos de una partición
squeue -t pd → ver trabajos en un estado (pending)
squeue --start → Ver fecha prevista para comenzar.
sacct -j 2454524 → ver estado de un trabajo
scontrol show job= 2454524 → estado de un trabajo
scontrol show job= 2454524 → estado de un trabajo, mas info
scontrol show node drago31010017 → ver estado del nodo
scontrol show partitions → ver particiones
sinfo -r → ver nodos con errores

PD → pendiente: Job esperando por asignación de recursos.
R → ejecución: Job tiene asignado recursos y se está ejecutando.
S → suspendido: Job ha sido suspendido su ejecución y se han liberado sus recursos para otros jobs.
CA → cancelado: Job fue explícitamente cancelado por usuario o administrador.
CG → completándose: Job está en proceso de completarse. Algunos procesos sobre algunos nodos pueden estar todavía activos
CD → completado: Job ha terminado todos los procesos sobre todos los nodos de forma correcta con una salida de 0.
F → fallido: Job terminó con un código de salida distinto a 0 y por tanto en fallo.

Directivas

```

#SBATCH -p generic # Partición (cola)
#SBATCH -N 1 # Numero de nodos
#SBATCH -n 1 # Numero de cores(CPUs)
#SBATCH --gres=gpu:1 # SOLO en partición GPU
#SBATCH --mem 100 # Bloque de memoria para todos los nodos
#SBATCH -t 0-02:00 # Duración (D-HH:MM)
#SBATCH -o slurm.%N,%j.out #STDOUT
#SBATCH -e slurm.%N,%j.err #STDERR
#SBATCH --mail-type=BEGIN,END,FAIL # Notificación cuando el trabajo empieza, termina o falla
#SBATCH --mail-user=micorreo@csic.es # Enviar correo a la dirección
  
```

Variables de entorno. SLURM

SLURM_JOB_NAME → Nombre del job
SLURM_JOB_ID → Id del job asignado
SLURM_CPUS_PER_TASK → Número de cpus solicitadas por tarea
SLURM_NTASKS → igual que anterior, -n, --ntasks
SLURM_SUBMIT_DIR → Directorio desde donde sbatch fue invocado.
SLURM_NODELIST → Listado de los nodos asignado al trabajo.
SLURM_NNODES → Número de nodos asignados al trabajo.
SLURM_TMPDIR → Directorio temporal del trabajo (scratch del sistema).

Módulos

module list → Muestra los módulos cargados
module av → Muestra los módulos disponibles
module spider → Muestra un módulo y sus dependencias
module load → Carga un modulo
module purge → Descarga todos los módulos cargados

Rutas

Home → /lustre/home/"centro"/"usuario"
Grupo → /lustre/home/"centro"/"grupoinvestigación"
Centro → /lustre/home/"centro"/

Almacenamiento Lustre

lfs df -h → ver espacio ocupado lustre
lfs quota -h \$HOME → Ver cuota de nuestro usuario y grupo en \$HOME
lfs find \$HOME -type f -name "README.md" → Busca ficheros que se llamen "README.md"
lfs find \$HOME -type d -name "formacion" → Busca directorio que se llamen "formacion"
lfs find \$HOME -size +2M → Busca ficheros mayores de 2megas
lfs find \$HOME -mtime +7 → Busca archivos de hace mas de 7 días
lfs getstripe \$HOME → Muestra el stripe del directorio

Información y contacto

<https://aic.csic.es> → Pagina general del Área de AIC (Área de informática científica)
<https://docaic.csic.es> → Pagina de documentación de AIC
soportheastic@csic.es → Correo de peticiones e incidencias
aic@csic.es → Correo general del área

Comandos básicos Linux

pwd → Muestra la ruta actual de trabajo
ls → Lista contenido del directorio actual o del que le indiquemos
cd → Cambiar de directorio
mkdir → Crea una carpeta
rmdir → Borra una carpeta
touch → Crea un archivo vacío
rm → Borra un fichero
man → Saca el manual.

mv → Mueve un fichero
cp → Copia el fichero
wget "url" → Descarga el fichero de la url indicada
ps --aux → muestra los procesos en ejecución
htop → muestra los procesos en ejecución mas visual que ps
cat "fichero" → Muestra el contenido del fichero(texto)
vi → Editor de texto, en todo los Linux
nano → Editor de texto, mas moderno
scp → Copia ficheros entre servidores
rsync → Copia ficheros entre servidores (mejor que scp)

Ejemplos comandos

mv "fichero o patrón" "origen" "destino" → Mueve el fichero o resultado del patrón del origen al destino
cp "fichero o patrón" "origen" "destino" → Copia el fichero o resultado del patrón del origen al destino
cp -r "directorio patrón" "origen" "destino" → Copia el directorio o resultado del patrón recursivamente, del origen al destino
Scp →
rsync -a dir2/ usuario@drago.csic.es:/lustre/home/centro1/usuario/dir2 → Mover datos desde pc/portátil a home de usuario en Drago
rsync -aP dir2/ usuario@drago.csic.es:/lustre/home/centro1/usuario/dir2 → Transferir datos reanudando envío tras interrupción de la misma
scp dir2/ usuario@drago.csic.es:/lustre/home/centro1/usuario/dir2 → copia dir2 de tu equipo local, a la ruta de drago

RSA

ssh-keygen -t rsa -C usuario@drago -f /home/usuario/.ssh/id_rsa_usuario_drago → Generar RSA
ssh-copy-id -i rutaDeLaRSA/id_rsa.pub usuario@drago.csic.es → Copia de la RSA a drago

Git

Configuración inicial

git config --global http.sslVerify false
git config --global user.email "mimail@csic.es"
git config --global user.name "minombre"
git clone https://rutadelbranch.git

Branchs

Scripts slurm
<https://git.csic.es/area-de-informatica-cientifica/slurm-scripts.git>

git status → Muestra diferencias entre directorio local y remoto
git fetch -v → Actualiza listado de cambios entre directorio local y remoto
git pull → Actualiza el directorio local a partir del remoto

Formación

<https://git.csic.es/area-de-informatica-cientifica/formacion.git>

Contenedores

<https://git.csic.es/area-de-informatica-cientifica/drago-hpc-containers.git>

Charliecloud

ch-image list → Lista las imágenes, si no tienes fichero /var/tmp/"usuario".ch lo crea
ch-image build --force . →(en la ruta del fichero docker, que se tiene que llamar Dockerfile) Lo descarga a /var/tmp
ch-image list → para ver que lo ha creado
ch-convert "Imagen" "directorio destino" →
ch-convert rockylinux8 /tmp/\$USER/rocky8 → desempaquetar la imagen rockylinux8 (directorio donde está el docker) en el directorio /tmp/\$USER/rocky8 El directorio /tmp/\$USER tiene que existir
ch-run /tmp/\$USER/rocky8 -- echo hola → corre el comando echo hola en el contenedor que hemos puesto en /tmp/\$USER/rocky8
ch-run -w /tmp/\$USER/rocky8 -- echo hola → como el anterior, cargando el contenedor en modo escritura
ch-run /tmp/\$USER/rocky8 -- echo hola -b /lustre/home/admins/ → como el anterior, pero solo deja escribir en la ruta que le pasas

Información y contacto

www.aic.csic.es → Pagina general del Área de AIC (Área de informática científica)
www.docaic.csic.es → Pagina de documentación de AIC
soportheastic@csic.es → Correo de peticiones e incidencias
aic@csic.es → Correo general del área