

Job Options	Job Options (cont)	Job submission
Account to be charged for resources used <code>-A, --acco unt =<a cco unt></code>	E-mail notification type (e.g., begin, end, fail, requeue, all) <code>--mail -ty pe= <ty pe></code>	<code>sbatch</code> Submit a batch script
Job array specification (sbatch only) <code>-a, --arra y=< ind ex></code>	E-mail address <code>--mail -us er= <ad dre ss></code>	<code>salloc</code> Request allocation for interactive job
Initiate job after specified time <code>-b, --begi n=< tim e></code>	Memory required per allocated node (e.g., 16GB) <code>--mem= <si ze> [units]</code>	<code>srun</code> Request allocation and run an application
Required node features <code>-C, --cons tra int =<f eat ure s></code>	Memory required per allocated CPU (e.g., 2GB) <code>--mem- per -cp u=< siz e>[uni ts]</code>	sbatch and salloc examples
Bind tasks to specific CPUs (srun only) <code>--cpu-bind=<type></code>	Specify host names to include in job allocation <code>-w, --node lis t=< hos tna mes></code>	<code># Request interactive job on debug node with 4 CPUs</code> <code>salloc -p standby -c 4</code> <code># Request intera ctive job with V100 GPU</code> <code>salloc -p comm_g pu_ inter --ntasks=1 --gpus=3</code> <code># Submit batch job</code> <code>sbatch runjob.slurm</code>
Number of CPUs required per task <code>-c, --cpus -pe r-t ask =<c oun t></code>	Number of nodes required for the job <code>-N, --node s=< cou nt></code>	sprio options
Defer job until specified jobs reach specified state <code>-d, --depe nde ncy =<s tat e:j obi d></code>	Number of tasks to be launched <code>-n, --ntas ks= <co unt></code>	Output format to display <code>-o, --form at= <op tio ns></code>
Specify distribution methods for remote processes <code>-m, --dist rib uti on= <me tho d[: m et hod]></code>	Number of tasks to be launched per node <code>--ntas ks- per -no de= <co unt></code>	Filter by job IDs (csl) <code>`j, --jobs=<job_id_list>`</code>
File in which to store job error messages (sbatch and srun only) <code>-e, --erro r=< fil ena me></code>	File in which to store job output (sbatch and srun only) <code>-o, --outp ut= <fi lena me></code>	Show more available information <code>-l, --long</code>
Specify host names to exclude from job allocation <code>-x, --excl ude =<n ame></code>	Partition in which to run the job <code>-p, --part iti on= <na mes></code>	Show the normalized priority factors <code>-n, --norm</code>
Reserve all CPUs and GPUs on allocated nodes <code>--excl usive</code>	Signal job when approaching time limit <code>--sign al= [B:]<n um> [@time]</code>	Filter by partitions (csl) <code>-p, --part iti on= <pa rti tio n _l ist></code>
Export specified environment variables (e.g., all, none) <code>--expo rt= <na me= val ue></code>	Limit for job run time <code>-t, --time =<t ime></code>	Filter by users (csl) <code>-u, --user =<u ser _li st></code>
Number of GPUs required per task <code>--gpus -pe r-t ask =<l ist></code>	The options can be used on the command line or in the case of a submission script they must be preceed by #SBATCH	csl = comma-separated list
Job name <code>-J, --job- nam e=< nam e></code>		
Prepend task ID to output (srun only) <code>-l, --label</code>		



By **Guillermo (guilleaf)**
cheatography.com/guilleaf/
bit.ly/guilleaf

Not published yet.
 Last updated 28th March, 2024.
 Page 1 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

sprio examples

```
# View normalized job priorities
for your own jobs
sprio -nu $USER

# View normalized job priorities
for specified partition
sprio -nlp standby
```

scancel examples

```
# Cancel specific job
scancel 314159

# Cancel all your own jobs
scancel -u $USER

# Cancel your own jobs on
specified partition
scancel -u $USER -p standby

# Cancel your own jobs in
specified state
scancel -u $USER -t pending
```

scancel options

Restrict to the specified account

```
-A, --account=<account_list>
```

Restrict to jobs with specified name

```
-n, --name=<job_name>
```

Restrict to jobs using the specified host names (csl)

```
-w, --node list=<hostnames>
>
```

Restrict to the specified partition

```
-p, --partition=<partition>
n>
```

Restrict to the specified user

```
-u, --user=<username>
```

csl = comma-separated list

squeue examples

```
# View your own job queue with
estimated start times
squeue --me

# View own job queue with
estimated start times for
pending jobs
squeue --me --start

# View job queue on specified
partition in long format
squeue -lp epyc-64
```

squeue options

Filter by accounts (csl)

```
-A, --account=<account_list>
```

Output format to display

```
-o, --format=<options>
```

Filter by job IDs (csl)

```
-j, --jobs=<job_id_list>
```

Show more available information

```
-l, --long
```

Filter by your own jobs

```
--me
```

Filter by job names (csl)

```
-n, --name=<job_name_list>
```

Filter by partitions (csl)

```
-p, --partition=<partition_list>
```

Sort jobs by priority

```
-P, --priority
```

Show the expected start time and resources to be allocated for pending jobs

```
--start
```

Filter by states (csl)

```
-t, --states=<state_list>
```

Filter by users (csl)

```
-u, --user=<user_list>
```

csl = comma-separated list

Job Management

squeue View information about jobs in queue

scancel Signal or cancel jobs, job arrays, or job steps

sprio View job scheduling priorities

Partition and node information

sinfo View information about nodes and partitions

scontrol View or modify configuration and state

sinfo options

Output format to display

```
-o, --format=<options>
```

Show more available information

```
-l, --long
```

Show information in a node-oriented format

```
-N, --Node
```

Filter by host names (comma-separated list)

```
-n, --nodes=<hostnames>
```

Filter by partitions (comma-separated list)

```
-p, --partition=<partition_list>
_l list>
```

Filter by node states (comma-separated list)

```
-t, --states=<state_list>
```

Show summary information

```
-s, --summarize
```

sinfo examples

```
# View all partitions and nodes
by state
```

```
sinfo
```

```
# Summarize node states by
partition
```

```
sinfo -s
```

```
# View nodes in idle state
```

```
sinfo --states=idle
```



By Guillermo (guilleaf)
cheatography.com/guilleaf/
bit.ly/guilleaf

Not published yet.
 Last updated 28th March, 2024.
 Page 2 of 3.

Sponsored by CrosswordCheats.com
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

sinfo examples (cont)

```
> # View nodes for specified partition in
long, node-oriented format
sinfo -lNp standby
```

scontrol actions and options

Show more details

```
-d, --details
```

Show information on one line

```
-o, --oneliner
```

Show partition

```
show partition <pa rti tio n>
```

Show node

```
show node <ho stn ame>
```

Show job

```
show job <jo b_i d>
```

Hold Jobs

```
hold <jo b_l ist>
```

Release Jobs

```
release <jo b_l ist>
```

Show Hostnames

```
show hostnames
```

scontrol examples

```
# View information for specified
partition
scontrol show partition standby
# View inform ation for
specified node
scontrol show node tcocs002
# View detailed inform ation for
running job
scontrol show job 314159 -d
# View hostnames for job (one
name per line)
scontrol show hostnames
```

Slurm environment variables

Number of tasks in job array

```
SLURM_ ARR AY_ TAS K_COUNT
```

Job array task ID

```
SLURM_ ARR AY_ TASK_ID
```

Number of CPUs requested per task

```
SLURM_ CPU S_P ER_TASK
```

Account used for job

```
SLURM_ JOB _AC COUNT
```

Job ID

```
SLURM_ JOB_ID
```

Job Name

```
SLURM_ JOB _NAME
```

List of nodes allocated to job

```
SLURM_ JOB _NO DELIST
```

Number of nodes allocated to job

```
SLURM_ JOB _NU M_NODES
```

Partition used for job

```
SLURM_ JOB _PA RTITION
```

Number of job tasks

```
SLURM_ NTASKS
```

MPI rank of current process

```
SLURM_ PROCID
```

Directory from which job was submitted

```
SLURM_ SUB MIT_DIR
```

Number of job tasks per node

```
SLURM_ TAS KS_ PER _NODE
```

Slurm Environment Variables examples

```
# Specify OpenMP threads
export OMP_NU M_T HRE ADS =$S -
LUR M_C PUS _PE R_TASK
# Specify MPI tasks
srun -n $SLURM _NTASKS ./mpi_ -
program
```



By **Guillermo** (guilleaf)
cheatography.com/guilleaf/
bit.ly/guilleaf

Not published yet.
 Last updated 28th March, 2024.
 Page 3 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>