SLURM 101

Here's a simple Slurm job script:

```
$ cat slurm-job.sh
#!/usr/bin/env bash

#SBATCH -o slurm.sh.out
#SBATCH -p defq

echo "In the directory: `pwd`"
echo "As the user: `whoami`"
echo “write this is a file" > analysis.output
sleep 60
```

Submit the job:

```
$ module load slurm
$ sbatch slurm-job.sh
Submitted batch job 106
```

List jobs:

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
   106 defq     slurm-jo  rstober   R   0:04     1 node001
```

Get job details:

```
$ scontrol show job 106
JobId=106 Name=slurm-job.sh
  UserId=rstober(1001) GroupId=rstober(1001)
  Priority=4294901717 Account=(null) QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 ExitCode=0:0
  RunTime=00:00:07 TimeLimit=UNLIMITED TimeMin=N/A
  SubmitTime=2013-01-26T12:55:02 EligibleTime=2013-01-26T12:55:02
  StartTime=2013-01-26T12:55:02 EndTime=Unknown
  PreemptTime=None SuspendTime=None SecsPreSuspend=0
  Partition=defq AllocNode:Sid=atom-head1:3526
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=atom01
  BatchHost=atom01
  NumNodes=1 NumCPUs=2 CPUs/Task=1 ReqS:C:T=*:*:*
  MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) Gres=(null) Reservation=(null)
```

```
  Shared=0 Contiguous=0 Licenses=(null) Network=(null)
  Command=/home/rstober/slurm/local/slurm-job.sh
  WorkDir=/home/rstober/slurm/local
```

Suspend a job (root only):

```
# scontrol suspend 135
# squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
   135  defq simple.s  rstober  S   0:10   1    node001
```

Resume a job (root only):

```
# scontrol resume 135
# squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
   135  defq simple.s  rstober  R   0:13   1    node001
```

Kill a job. Users can kill their own jobs, root can kill any job.

```
$ scancel 135
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Hold a job:

```
$ squeue
 JOBID PARTITION     NAME     USER  ST      TIME  NODES NODELIST(REASON)
   139     defq   simple  rstober  PD      0:00      1 (Dependency)
   138     defq   simple  rstober   R      0:16      1 node001
$ scontrol hold 139
$ squeue
 JOBID PARTITION     NAME     USER  ST      TIME  NODES NODELIST(REASON)
   139     defq   simple  rstober  PD      0:00      1 (JobHeldUser)
   138     defq   simple  rstober   R      0:32      1 node001
```

Release a job:

```
$ scontrol release 139
$ squeue
 JOBID PARTITION     NAME     USER  ST      TIME  NODES NODELIST(REASON)
   139     defq   simple  rstober  PD      0:00      1 (Dependency)
   138     defq   simple  rstober   R      0:46      1 node001
```

List partitions:

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
defq*        up    4:00:00     20   idle node[001-020]
short        up 1-00:00:00     20   idle node[001-020]
long         up 20-00:40:0     20   idle node[001-020]
```

Submit a job that's dependant on a prerequisite job being completed:

Here's a simple job script. Note that the Slurm -J option is used to give the job a name.

```
#!/usr/bin/env bash

#SBATCH -p defq
#SBATCH -J simple

sleep 60
```

Submit the job

```
$ sbatch simple.sh
Submitted batch job 149
```

Now we'll submit another job that's dependent on the previous job. There are many ways to specify the dependency conditions, but the "singleton" is the simplest. The Slurm -d singleton argument tells Slurm not to dispatch this job until all previous jobs with the same name have completed.

```
$ sbatch -d singleton simple.sh
Submitted batch job 150

$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
   150 defq   simple  rstober  PD  0:00  1 (Dependency)
   149 defq   simple  rstober   R  0:17  1 node001
```

Once the prerequisite job finishes the dependent job is dispatched.

```
$ squeue
 JOBID PARTITION NAME USER ST TIME  NODES NODELIST(REASON)
   150 defq   simple  rstober   R   0:31  1 node001
```