

AT TEAM INSTANCE

Disabling slurmd

```
venkata@t-green:~$ sudo systemctl disable slurmd
venkata@t-green:~$ systemctl status -l slurmd
● slurmd.service - Slurm node daemon
   Loaded: loaded (/etc/systemd/system/slurmd.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
```

```
/a/apps/tb/slurm-green/22.05.6/etc/slurm.conf
```

```
venkata@t-green:~$ ls -al /a/apps/tb/slurm-green/22.05.6/etc/slurm.conf
-rw-r--r-- 1 root root 3247 Apr 12 13:43 /a/apps/tb/slurm-green/22.05.6/etc/slurm.conf
venkata@t-green:~$
```

```

# slurm.conf file generated by configurator.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.

ClusterName=p01
SlurmctldHost=t-green

#DisableRootJobs=NO
#EnforcePartLimits=NO
#EpiLogo
#EpiLogoSlurmctld
#FirstJobId=1

#--job info: MaxJobId=9763378
#SuspendRate
#SuspendTime
#
#
# COMPUTE NODES
#AutoDetectnvml

NodeName=i-hemanth,i-kush,i-roja,i-venkata Weight=20 State=UNKNOWN
PartitionName=i0 Nodes=i-hemanth,i-kush,i-roja,i-venkata Default=NO MaxTime=INFINITE State=UP

PartitionName=batch Nodes=ALL Default=YES MaxTime=INFINITE State=UP QOS=batch DefMemPerCPU=8000

# slurm.conf file generated by configurator.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.

ClusterName=p01
SlurmctldHost=t-sh01

#DisableRootJobs=NO
#EnforcePartLimits=NO
#EpiLogo
#EpiLogoSlurmctld
#FirstJobId=1

#--job info: MaxJobId=9763378
#SuspendRate
#SuspendTime
#
#
# COMPUTE NODES
#AutoDetectnvml

NodeName=i-yechang Weight=1 CPU=2 Boards=1 SocketsPerBoard=1 CoresPerSocket=1 ThreadsPerCore=2 RealMemory=10
PartitionName=i0 Nodes=i-blue,t-brown,t-green,t-red Weight=10 CPU=2 Boards=1 SocketsPerBoard=1 CoresPerSocket=1
NodeName=i-akshata,i-dhruv,i-harshila,i-hemanth,i-kush,i-mayur,i-mandikonda,i-pawan,i-pranod,i-prem,i-raja,t-i

PartitionName=c0 Nodes=i-yechang Default=YES MaxTime=INFINITE State=UP
PartitionName=i0 Nodes=i-blue,t-blue,t-brown,t-green,t-red Default=NO MaxTime=INFINITE State=UP
PartitionName=i10 Nodes=i-akshata,i-dhruv,i-harshila,i-hemanth,i-kush,i-mayur,i-mandikonda,i-pawan,i-pranod,i-
PartitionName=c0 Nodes=i-yechang Default=YES MaxTime=INFINITE State=UP
PartitionName=batch Nodes=ALL Default=YES MaxTime=INFINITE State=UP QOS=batch DefMemPerCPU=8000
#DefMemPerCPU=8000

```

1. ClusterName=p01
2. SlurmctlHost=t-green
3. NodeName=i-hemanth,i-kush,i-roja,i-venkata Weight=20 State=UNKNOWN
4. PartitionName=i0 Nodes=i-hemanth,i-kush,i-roja,i-venkata Default=NO MaxTime

We are initializing slurmctld from slurm-green instance located at the path. We are updating the path in for the service to point to slurm-green controller along with passing the configuration file.

```
sudo vi /etc/systemd/system/slurmctld.service
```

```
/a/apps/tb/slurm-green/22.05.6/sbin/slurmctld -D -c -f /a/apps/tb/slurm-green/22.05.6/etc/slurm.conf
```

```
venkata@t-green:~$ sudo cat /etc/systemd/system/slurmctld.service
[Unit]
Description=Slurm controller daemon
After=network-online.target munge.service
ConditionPathExists=/a/apps/tb/slurm-green/22.05.6/etc/slurm.conf

[Service]
Type=simple
EnvironmentFile=-/etc/sysconfig/slurmctld
ExecStart=/a/apps/tb/slurm-green/22.05.6/sbin/slurmctld -D -c -f /a/apps/tb/slurm-green/22.05.6/etc/slurm.conf $SLURMCTLD_OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
LimitNOFILE=562930
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
venkata@t-green:~$
```

Enabling & starting the slurmctld

```
sudo systemctl enable slurmctld
```

```
sudo systemctl start slurmctld
```

```
venkata@t-green:~$ sudo systemctl enable slurmctld
venkata@t-green:~$ sudo systemctl start slurmctld
venkata@t-green:~$
```

Service running can be validated using the command

```
systemctl status -l slurmctld
```

```
venkata@t-green:~$ systemctl status -l slurmctld
● slurmctld.service - Slurm controller daemon
   Loaded: loaded (/etc/systemd/system/slurmctld.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-04-13 18:43:46 PDT; 43min ago
     Main PID: 1898 (slurmctld)
        Tasks: 10 (limit: 2307)
       Memory: 10.3M
      CGroup: /system.slice/slurmctld.service
              └─1898 /a/apps/tb/slurm-green/22.05.6/sbin/slurmctld -D -c -f /a/apps/tb/slurm-green/22.05.6/etc/slurm.conf
                 1899 slurmctld: slurmscriptd
venkata@t-green:~$
```

Connected nodes can be viewed using as follows

```
sinfo
```

```
venkata@t-green:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite     4    idle i-hemanth,i-kush,i-roja,i-venkata
batch*     up    infinite     4    idle i-hemanth,i-kush,i-roja,i-venkata
venkata@t-green:~$
```

The output shows that there are two partitions available: i0 and batch.

Both partitions have 4 nodes in an idle state.

The nodes are

1. i-hemanth
2. i-kush
3. i-roja
4. i-venkata

2. Configure ind WS as the Slurm work node of team cluster p01, refer to class cluster p01 configuration. The successful result should have sinfo on team node to report the team member ind. WS

AT INDIVIDUAL INSTANCE

Updating the bash default scripts to point to slurm-green files

```
sudo vi /etc/profile.d/slurm.sh
```

Adding the following line to the above file to override pointing to slurm-green path in individual instances

```
PATH=/a/apps/tb/slurm-green/22.05.6/bin:/a/apps/tb/slurm-green/22.05.6/sbin:$PATH
```

```
venkata@i-venkata:~$ sudo cat /etc/profile.d/slurm.sh
PATH=/a/apps/tb/slurm-green/22.05.6/bin:/a/apps/tb/slurm-green/22.05.6/sbin:$PATH
PATH=$PATH:/a/apps/gb/slurm/22.05.6/bin:/a/apps/gb/slurm/22.05.6/sbin
venkata@i-venkata:~$
```

Restart the slurmd daemon on individual server using following command

```
sudo systemctl restart slurmd
sudo systemctl status slurmd
```

```
saiteja — venkata@i-venkata: ~ — ssh venkata@ec2-3-215-202-17.compute-1.amazonaws.com
venkata@i-venkata:~$ sudo systemctl restart slurmd
venkata@i-venkata:~$ sudo systemctl status slurmd
● slurmd.service - Slurm node daemon
   Loaded: loaded (/etc/systemd/system/slurmd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-04-13 19:34:36 PDT; 8s ago
     Main PID: 56018 (slurmd)
       Tasks: 1 (limit: 2307)
      Memory: 1.1M
    CGroup: /system.slice/slurmd.service
```

Restart the bash and run the following command to show slurm controller information

```
sinfo
```

```
venkata@i-venkata:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite     4    idle i-hemanth,i-kush,i-roja,i-venkata
batch*     up    infinite     4    idle i-hemanth,i-kush,i-roja,i-venkata
venkata@i-venkata:~$
```

The config is shown by running the following command

```
scontrol show config
```

The config output is shown at the end of config stating the t-green cluster controller is UP

```
Slurmctld(primary) at t-green is UP
```

```
venkata@i-venkata: /home/v
Cgroup Support Configuration:
AllowedKmemSpace      = (null)
AllowedRAMSpace       = 100.0%
AllowedSwapSpace      = 0.0%
CgroupAutomount       = no
CgroupMountpoint      = (null)
CgroupPlugin          = (null)
ConstrainCores        = no
ConstrainDevices      = no
ConstrainKmemSpace    = no
ConstrainRAMSpace     = no
ConstrainSwapSpace    = no
IgnoreSystemd         = no
IgnoreSystemdOnFailure = no
MaxKmemPercent        = 100.0%
MaxRAMPercent         = 100.0%
MaxSwapPercent        = 100.0%
MemorySwappiness      = (null)
MinKmemSpace          = 30 MB
MinRAMSpace           = 30 MB

Slurmetld(primary) at t-green is UP
venkata@i-venkata:/home/venkata$
```

TESTING & VALIDATION

We are running the `srun/sbatch` command from `t-green` instance to schedule a job to be delegated to worker nodes.

SRUN

We are using the following commands(`srun`) to trigger a new job to all the worker nodes

```
srun -p i0 --time=1:0:0 --ntasks=4 --mem=1mb --pty bash --nodes=4
```

```
saiteja — venkata@t-green: ~ — ssh venkata@ec2-3-215-202-17.compute-1.amazonaws.com — 162x25
venkata@t-green:~$ srun -p i0 --time=1:0:0 --ntasks=4 --mem=1mb --pty bash --nodes=4
srun: job 181 queued and waiting for resources
srun: job 181 has been allocated resources
```

Status of the job can be seen using `squeue`

```
squeue --long
sinfo
```

Since the job is in **COMPLETING** state, we can see the node status using `sinfo`. The State also is updated to **COMP**

```
venkata@i-venkata:~$ squeue --long
Thu Apr 13 19:45:48 2023
  JOBID PARTITION   NAME   USER  STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
   181      i0      bash  venkata COMPLETI    0:00   1:00:00     4 i-hemanth,i-kush,i-roja,i-venkata
venkata@i-venkata:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0        up    infinite     4    comp i-hemanth,i-kush,i-roja,i-venkata
batch*    up    infinite     4    comp i-hemanth,i-kush,i-roja,i-venkata
venkata@i-venkata:~$
```

Once the job is completed, there are no more jobs in the queue and nodes are set to idle state as show below to process the next jobs

```
venkata@i-venkata:~$ squeue --long
Thu Apr 13 19:46:50 2023
      JOBID PARTITION     NAME     USER      STATE      TIME TIME_LIMI  NODES NODELIST(REASON)
venkata@i-venkata:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite      4    idle i-hemanth,i-kush,i-roja,i-venkata
batch*    up    infinite      4    idle i-hemanth,i-kush,i-roja,i-venkata
venkata@i-venkata:~$
```

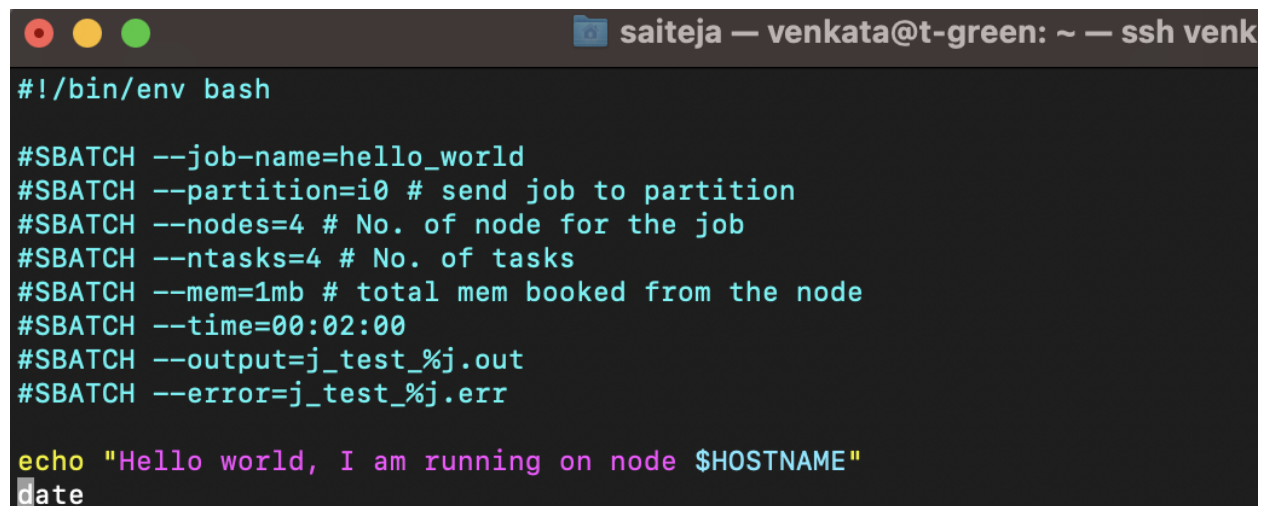
SBATCH

Executing a batch file using sbatch command. We are creating a new hello world file that is going to print hello world to the output from all the nodes. Alongside we are configuring some SBATCH properties. The file has the following content.

```
vi helloworld
```

Add the following content

```
#!/bin/env bash
#SBATCH --job-name=hello_world
#SBATCH --partition=i0 # send job to partition
#SBATCH --nodes=4 # No. of node for the job
#SBATCH --ntasks=4 # No. of tasks
#SBATCH --mem=1mb # total mem booked from the node
#SBATCH --time=00:02:00
#SBATCH --output=j_test_%j.out
#SBATCH --error=j_test_%j.err
echo "Hello world, I am running on node $HOSTNAME"
date
```

A terminal window titled "saiteja — venkata@t-green: ~ — ssh venk" displays the content of the helloworld script. The script is a bash file that uses the sbatch command to submit a job to the i0 partition. It specifies 4 nodes, 4 tasks, 1mb memory, and a 2-minute time limit. The output and error files are set to j_test_%j.out and j_test_%j.err respectively. The script also includes an echo statement to print "Hello world, I am running on node \$HOSTNAME" and a date command.

```
#!/bin/env bash

#SBATCH --job-name=hello_world
#SBATCH --partition=i0 # send job to partition
#SBATCH --nodes=4 # No. of node for the job
#SBATCH --ntasks=4 # No. of tasks
#SBATCH --mem=1mb # total mem booked from the node
#SBATCH --time=00:02:00
#SBATCH --output=j_test_%j.out
#SBATCH --error=j_test_%j.err

echo "Hello world, I am running on node $HOSTNAME"
date
```


We are using the following commands(sbatch) to trigger a new job to all the worker nodes

```
sbatch helloworld
```

```
venkata@t-green:~$ sbatch helloworld
Submitted batch job 182
venkata@t-green:~$
```

Status of the job can be seen using squeue

```
squeue --long
sinfo
```

Since the job is in **RUNNING** state, we can see the node status using sinfo. The individual daemon state also is updated to **ALLOC**

```
venkata@t-green:~$ squeue --long
Thu Apr 13 21:00:52 2023
      JOBID PARTITION     NAME     USER  STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
      191      i0 hello_wo  venkata  RUNNING    0:21    2:00      4 i-hemanth,i-kush,i-roja,i-venkata
venkata@t-green:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite      4   alloc i-hemanth,i-kush,i-roja,i-venkata
batch*    up    infinite      4   alloc i-hemanth,i-kush,i-roja,i-venkata
venkata@t-green:~$
```

Since the job is in **COMPLETING** state, we can see the node status using sinfo. The State also is updated to **COMP**

```
venkata@t-green:~$ squeue --long
Thu Apr 13 21:02:44 2023
      JOBID PARTITION     NAME     USER  STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
      191      i0 hello_wo  venkata  COMPLETI    2:10    2:00      3 i-hemanth,i-kush,i-roja
venkata@t-green:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite      3   comp i-hemanth,i-kush,i-roja
i0         up    infinite      1   idle i-venkata
batch*    up    infinite      3   comp i-hemanth,i-kush,i-roja
batch*    up    infinite      1   idle i-venkata
venkata@t-green:~$
```

Once the job is completed, there are no more jobs in the squeue and nodes are set to idle state as show below to process the next jobs

```
venkata@t-green:~$ squeue --long
Thu Apr 13 21:03:44 2023
      JOBID PARTITION     NAME     USER  STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
venkata@t-green:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
i0         up    infinite      4   idle i-hemanth,i-kush,i-roja,i-venkata
batch*    up    infinite      4   idle i-hemanth,i-kush,i-roja,i-venkata
venkata@t-green:~$
```

OUTPUT is as follows since we are using sbatch, only the first output is captured to file

```
venkata@t-green:~$ cat j_test_191.  
j_test_191.err j_test_191.out  
venkata@t-green:~$ cat j_test_191.err  
venkata@t-green:~$ cat j_test_191.out  
Hello world, I am running on node i-hemanth  
Fri Apr 14 04:00:31 UTC 2023  
venkata@t-green:~$
```

The job can be seen using sacct

```
sacct -X -u venkata --starttime="2023-04-14T04:00:00"  
--format=user,jobid,jobname,Partition,NodeList,ntasks,reqtres%40,submit,start,end,elapsed,TimeLimit,ReqMem,MaxRSS,Reason,WorkDir%20
```

```
venkata@i-hemanth:~$ sacct -X -u venkata --starttime="2023-04-14T04:00:00" --format=user,jobid,jobname,Partition,NodeList,ntasks,reqtres%40,submit,start,end,elapsed,TimeLimit,ReqMem,MaxRSS,Reason,WorkDir%20  
User JobID JobName Partition NodeList Ntasks ReqTRES Submit Start End Elapsed TimeLimit ReqMem MaxRSS Reason WorkDir  
venkata 191 hello_wor i i-hemanth,i-kue billing=4,cpu=4,mem=4M,node= 2023-04-14T04:00:31 2023-04-14T04:00:31 2023-04-14T04:02:41 00:02:10 00:02:00 AM None /home/venkata  
venkata@i-hemanth:~$
```


3. Update project report at wiki

This has been updated to reflect the changes that we've done above.

4. Contribution to the team progress

The team worked on setting up a Slurm controller and daemons and configuring it for the project.

Venkata and Kush took the lead on this task, handling the controller's installation and setup. They were responsible for ensuring that the configuration was accurate and that the controller was running smoothly.

Once the controller was set up, Hemanth and Roja took over to test and validate the entire process. They were responsible for ensuring that the controller was functioning correctly and that the system was working as intended. This involved testing various configurations and settings to ensure that the system was reliable and accurate.

Everyone setup their individual instances to work with slurm controller. While each team member had their specific responsibilities, everyone collaborated closely to ensure the success of the project. There were a lot of trial and error involved in setting up the controller and testing the process, and each team member contributed to this process. By working together, the team was able to achieve their goal and make progress towards the project's overall success.