1) Serial Job - basics
   a) Submit a serial job that:
      i) Is a serial (1 core) job
      ii) Emails you when is starts, ends and aborts
      iii) Has a maximum wall time of 2 minutes
      iv) Runs the 'hostname' command
   b) Make a note of the jobid when your job is submitted
   c) Watch your job run with the following command:
      i) "squeue -u $USER"
   d) Did you get the result emailed to your account
   e) Display the job output file.
   f) Examine the emails you have received.

2) Serial Job - output
   a) Submit a serial job that:
      i) As in the previous question
         - Is a serial (1 core) job
         - Emails you when is starts, ends and aborts
         - Has a maximum wall time of 2 minutes
         - Runs the 'hostname' command
      ii) Writes the standard output and standard error into separate files.
   b) Make a note of the jobid when your job is submitted
   c) Watch your job run with the following command:
      i) "squeue -u $USER"
      ii) "scontrol show job <jobid>"
   d) Did you get the result emailed to your account
   e) Display the job output file.
   f) Examine the emails you have received.

3) Serial Job - error
   a) Submit a serial job that:
      i) As in the previous question
         - Is a serial (1 core) job
         - Emails you when is starts, ends and aborts
         - Has a maximum wall time of 2 minutes
         - Runs the 'hostname' command
         - Writes the standard output and standard error into separate files.
      ii) Runs the non existent command 'hello'
   b) Make a note of the jobid when your job is submitted
   c) Watch your job run with the following command:
      i) "squeue -u $USER"
      ii) "sscontrol show job <jobid>""
   d) Did you get the result emailed to your account
   e) Display the job output and error files.
   f) Examine the emails you have received.

4) Serial Job - Walltime
   a) Submit a serial job that:
      i) As in question 2
         - Is a serial (1 core) job
         - Emails you when is starts, ends and aborts
         - Has a maximum wall time of 2 minutes
         - Runs the 'hostname' command
         - Writes the standard output and standard error into separate files.
      ii) Sleeps for 200 seconds
   b) Think about what will you think happen when this jobs runs?
      i) "squeue -u $USER"
      ii) The job output and error files.
      iii) The emails.
   c) Make a note of the jobid when your job is submitted
   d) Watch your job run with the following command:
      i) "squeue -u $USER"
      ii) "scontrol show job <jobid>"
   e) Did you get the result emailed to your account
   f) Examine the emails you have received.
   g) Display the job output and error files.

5) Serial Job – Job names
   a) Submit a serial job that:
      i) As in question 2
         - Is a serial (1 core) job
         - Emails you when is starts, ends and aborts
         - Has a maximum wall time of 2 minutes
         - Runs the 'hostname' command
         - Writes the standard output and standard error into separate files.
      ii) Sleeps for 30 seconds
      iii) Is named "my-named-job"

   b) Look at your job in the output of the following commands
      i) "squeue -u $USER"
      ii) "scontrol show job <jobid>"

   c) Look at the following job outputs:
      i) The job output and error files.
      ii) The emails

6) Interactive serial Job
   a) Open a second ssh session/terminal to the workshop cluster

   b) In the second ssh session submit a job that:
      i) Is a serial (1 core) job
      ii) Has a maximum wall time of 20 minutes
      iii) Emails you when the job is aborted, before it runs and a after it ends
      iv) Is named "my-first-interactive-job"
      v) Is interactive

   c) Wait for the job to be allocated after it is allocated answer the following questions
      i) Look at the command line you are on.
      ii) Which node are you on?
          Hint: "hostname"
      iii) Print and look at all the slurm variables.
          Hint: "printenv | grep SLURM"
      iv) Find out on which node is your job allocated.
          Hint: "echo $SLURM_NODELIST"
      v) What is the jobs name?
          Hint: "echo $SLURM_JOB_NAME"
      vi) In which directory are you?
          Hint: "pwd" command
      vii) Which directory has the job been submitted from?
          Hint: "echo $SLURM_SUBMIT_DIR"
      viii) What is the path to executable for this job?
          Hint: "echo $PATH"

   d) Open a shell on the inside of the job running in the allocated resources.
      Hint: "srun --pty -p interact bash"

e) Answer these basic questions
   i) Look at the command line you are on.
   ii) Which node are you on?
      Hint: “hostname”
   iii) Print and look at all the slurm variables.
      Hint: “printenv | grep SLURM”
   iv) Find out on which node is your job allocated.
      Hint: “echo $SLURM_NODELIST”
   v) What is the jobs name?
      Hint: “echo $SLURM_JOB_NAME”
   vi) In which directory are you?
      Hint: “pwd” command
   vii) Which directory has the job been submitted from?
      Hint: “echo $SLURM_SUBMIT_DIR”
   viii) What is the path to executable for this job?
      Hint: “echo $PATH”

f) From your first terminal on the login node look at your job in the output of the following commands:
   i) “squeue -u $USER”
   ii) “scontrol show job <jobid>”

g) Go back to your second terminal session

h) The current interactive job only uses 1 process, on 1 core, inside 1 task, inside 1 step, and does not use arrays but for future comparison with other more complex job types run the following commands and write down the results
   i) Give a list of node names where each process (there is only one in this case) of this job runs?
      Hint: "echo $SLURM_JOB_NODELIST"
   ii) On how many nodes (there is only one in this case) does this job run?
      Hint: "echo $SLURM_JOB_NUM_NODES"
      Hint: "echo $SLURM_NNODES"
   iii) What is the total number of tasks in this allocation
      Hint: "echo $SLURM_NTASKS"
      Hint: "echo $SLURM_NPROCS"
   iv) What is the number of tasks per node (listed by node)
      Hint: "echo $SLURM_TASKS_PER_NODE"
   v) How many steps are in this job
      Hint: "echo $SLURM_STEP_NUM_TASKS"
   vi) What is the ID of the current step
      Hint: "echo $SLURM_ SLURM_STEPID"
   vii) What is the array ID of this job?
      Hint: "echo $SLURM_ARRAY_TASK_ID"
      Answer: In this case there is none.

i) Optional for advanced Unix users: In the interactive job
   i) (Advanced Unix) List the jobs cpuset/cgroup. A cpuset or cgroup is used to assigning a set of processor and memory to a set of processes, and can be used by the scheduling system to keep a job from using resources assigned to other jobs.
      Hint: "ls /sys/fs/cgroup/cpuset/slurm/uid_$SLURM_JOB_UID/job_$SLURM_JOB_ID/step_$SLURM_STEPID"

   ii) (Advanced Unix) Verify that your current shell's process id is inside the cpuset.
      Hint: "echo $$" to find the process id of your current shell
      Hint: "cat /sys/fs/cgroup/cpuset/slurm/uid_$SLURM_JOB_UID/job_$SLURM_JOB_ID/step_$SLURM_STEPID/tasks"
      to list processes in the jobs cpuset.
   iii) (Advanced Unix) Show which cores and which memory set that your job is running on?
      Hint: "cat /sys/fs/cgroup/cpuset/slurm/uid_$SLURM_JOB_UID/job_$SLURM_JOB_ID/step_$SLURM_STEPID/cpuset.cpus"
         for core numbers that the job processes are running on.
      Hint: "cat /sys/fs/cgroup/cpuset/slurm/uid_$SLURM_JOB_UID/job_$SLURM_JOB_ID/step_$SLURM_STEPID/ cpuset.mems"
         for memory locatio(s) that the job memory may run on

j) Close the interactive shell by running the exit command

k) Open a shell on the inside of the job running in the allocated resources.
   Hint: "srun --pty -p interact bash"

l) What is the ID of the current step
   Hint: "echo $SLURM_ SLURM_STEPID"

m) Close the interactive shell by running the exit command
n) Close your allocated job by running the exit command

7) Job Arrays
   a) Submit a serial job array that:
      i) Has a maximum wall time of 2 minutes
      ii) Sleeps 30 seconds
      iii) Runs the command “hostname”
      iv) Is named “my-array-job”
      v) Has 12 tasks
      vi) Writes a output file to “slurm-q7_<jobid>_<arrayid>.out
      vii) Writes a error file to “slurm-q7_<jobid>_<arrayid>.err

   b) Run the following commands to see your job running
      i) “squeue –u $USER”
      ii) “scontrol show job <jobid>”

   c) Look at the job output files

8) Job Arrays
   a) Submit a serial job array
      i) Has a maximum wall time of 2 minutes
      ii) Sleeps 30 seconds
      iii) Runs the command “hostname”
      iv) Is named “my-array-job2”
      v) Has 12 tasks
      vi) Writes a output file to “slurm-q7_<jobid>_<arrayid>.out
      vii) Writes a error file to “slurm-q7_<jobid>_<arrayid>.err
      viii) Runs at most 2 jobs at once

   b) Run the following commands to see your job running
      i) “squeue –u $USER”
      ii) “scontrol show job <jobid>”

   c) Look at the job output files

9) Job Arrays
   a) Submit a serial job array
      i) Has a maximum wall time of 2 minutes
      ii) Sleeps 30 seconds
      iii) Runs the command “hostname”
      iv) Is named “my-array-job3”
      v) Has 4 tasks with indexes of: 1, 2, 7, -13
      vi) Writes a output file to “slurm-q9_<jobid>_<arrayid>.out
      vii) Writes a error file to “slurm-q9_<jobid>_<arrayid>.err
      viii) Runs at most 2 jobs at once

   b) Run the following commands to see your job running
      i) “squeue –u $USER”
      ii) “scontrol show job <jobid>”

   c) Look at the job output files

10)Job arrays

a) Submit a job that:
   i) Has 2 tasks with indexes of: 1, 4
   ii) Asks for 1 core per task
   iii) Has a maximum wall time of 2 minutes
   iv) Emails you when the job is aborted, before it runs and a after it ends
   v) Is named "my-array-var-job"

b) Look at the PBS environment variables:
   i) Give a list of node names where each process of this job runs?
      Hint: "echo $SLURM_JOB_NODELIST"
   ii) On how many nodes (there is only one in this case) does this job run?
      Hint: "echo $SLURM_JOB_NUM_NODES"
      Hint: "echo $SLURM_NNODES"
   iii) What is the total number of tasks in this allocation.?
      Hint: "echo $SLURM_NTASKS"
      Hint: "echo $SLURM_NPROCS
   iv) What is the number of tasks per node (listed by node)?
      Hint: "echo $SLURM_TASKS_PER_NODE"
   v) How many steps are in this job?
      Hint: "echo $SLURM_STEP_NUM_TASKS"
   vi) What is the task id in the array of this job?
      Hint: "echo $SLURM_ARRAY_TASK_ID"
   vii) Prints all the SLURM variables
      Hint: "printenv | grep SLURM"

c) Run the following commands to see your job running
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"

d) Look at the job output files

11) Array job with inputs example

Working example of an array job taking input from a file (This is advanced example, It was included as a request from a prior workshop, we may not have time to write this in the session, in that case just look at and run the answer script.)

a) Submit a serial job array that reads from a single file and runs a job for each line in the input file.
   i) Is named "my-input-array-job"
   ii) Has 4 tasks with 1 procs each.
   iii) Emails you when your job is complete.
   iv) The file that is used as input is named: "input.array"
b) Have each job array output double the first number and adding the second.
c) Run the job and see the output
d) As advanced work if time permits see if you can output in a single file as opposed to many array files.

12)MPI Jobs

a) Submit the start-mpi.sh job
b) Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
   iii) "scontrol show jobid -dd <jobid>"

c) Note how long it took to run
d) Edit the start-mpi.sh script to user 4 processors
e) Submit the edited script
f) Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
   iii) "scontrol show jobid -dd <jobid>"

g) Please list which nodes and cores the job is running on or scheduled to run on and how long it took to run.

13) MPI Interactive Job

a) Submit a job
   i) Asks 4 processors
   ii) Has a walltime of 20 minutes
   iii) Is named "my-interactive-mpijob"
   iv) Is interactive
b) After the job starts look at the SLURM environment variables
   (run: "printenv | grep -i slurm")
   i) What is the jobs id?
   Hint: "echo $SLURM_JOB_ID"
   ii) On how many nodes does this job run?
   Hint: "echo $SLURM_NNODES"
   iii) On how many processors does this job run?
   Hint: "echo $SLURM_NPROCS"
   iv) On which node(s) is your job allocated?
   Hint: "echo $SLURM_JOB_NODELIST"
   v) On how many processors per node is the job allocated?
   Hint: "echo $SLURM_JOB_CPUS_PER_NODE"
   vi) On how many tasks per node is the job allocated?
   Hint: "echo $SLURM_TASKS_PER_NODE"

c) Run the command "srun hostname "
   i) How many lines of output do you get.
d) Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
e) Exit your job with the exit command

14) MPI Interactive Jobs part2
    a) Submit a job
        i) Asks for 4 processors on a single node
        ii) Has a walltime of 20 minutes
        iii) Is named “my-interactive-mpijob2”
        iv) Is interactive
    b) After the job starts look at the SLURM environment variables
        i) (run: “printenv | grep -i slurm”)
        ii) What is the jobs id?
            Hint: “echo $SLURM_JOB_ID”
        iii) On how many nodes does this job run?
            Hint: “echo $SLURM_NNODES”
        iv) On how many processors does this job run?
            Hint: “echo $SLURM_NPROCS”
        v) On which node(s) is your job allocated?
            Hint: “echo $SLURM_JOB_NODELIST”
        vi) On how many processors per node is the job allocated?
            Hint: “echo $SLURM_JOB_CPUS_PER_NODE”
        vii) On how many tasks per node is the job allocated?
            Hint: “echo $SLURM_TASKS_PER_NODE”
    c) Run the command “srun hostname ”
        i) How many lines of output do you get.
    d) Look at the job with the following commands:
        i) “squeue –u $USER”
        ii) “scontrol show job <jobid>”
    e) Exit your job with the exit command

15) OpenMP jobs

a) Submit a job
   i) Asking for 1 node with 12 cores
   ii) Has a maximum walltime of 20 minutes
   iii) Is named “my-interactive-openmpjob”
   iv) Is interactive

b) After the job starts look at the SLURM environment variables (run: “printenv | grep -i slurm”)
   i) What is the jobs id?
      Hint: “echo $SLURM_JOB_ID”
   ii) On how many nodes does this job run?
      Hint: “echo $SLURM_NNODES”
   iii) On how many processors does this job run?
      Hint: “echo $SLURM_NPROCS”
   iv) On which node(s) is your job allocated?
      Hint: “echo $SLURM_JOB_NODELIST”
   v) On how many processors per node is the job allocated?
      Hint: “echo $SLURM_JOB_CPUS_PER_NODE”
   vi) On how many tasks per node is the job allocated?
      Hint: “echo $SLURM_TASKS_PER_NODE”

c) Run the command “srun hostname ”
   i) How many lines of output do you get.

d) Look at the job with the following commands:
   i) “squeue –u $USER”
   ii) “scontrol show job <jobid>”

e) Exit your job with the exit command

16) Hybrid Interactive Jobs

a) Submit a job that:
   i) Asking for 3 nodes with 2 task per node and 5 cores per task.
   ii) Is named "my-interactive-hybridjob"
   iii) Has a maximum walltime of 20 minutes
b) After the job starts look at the SLURM environment variables (run: "printenv | grep -i slurm")
   i) What is the jobs id?
      Hint: "echo $SLURM_JOB_ID"
   ii) On how many nodes does this job run?
      Hint: "echo $SLURM_NNODES"
   iii) On how many processors does this job run?
      Hint: "echo $SLURM_NPROCS"
   iv) On which node(s) is your job allocated?
      Hint: "echo $SLURM_JOB_NODELIST"
   v) On how many processors per node is the job allocated?
      Hint: "echo $SLURM_JOB_CPUS_PER_NODE"
   vi) On how many tasks per node is the job allocated?
      Hint: "echo $SLURM_TASKS_PER_NODE"
c) Run the command "srun hostname "
   i) How many lines of output do you get.
d) Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
e) Please list which nodes and cores the job is running on or scheduled to run on.
f) Exit your job with the exit command

17) Jobs and memory

a) Take the start-mem.pbs script and edit it so that is asks for: --mem-per-cpu=12000
b) Submit a job from the script you edited . Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
c) How much memory does this job use?

18)Jobs and memory

a) Take the start-mem.pbs script and edit it so that is asks for: --mem=12000
b) Submit a job from the script you edited . Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
c) How much memory does this job use?

19)Jobs and memory (mem,pmem)

a) Take the start-mem.pbs script and edit it so that is asks for:
   --mem-per-cpu =3000mb
b) Submit a job from the script you edited . Look at the job with the following commands:
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
c) How much memory does this job use?

20) Jobs and memory (appropriate resources)

a) Create a job run the "cryptic" program edit the start-mem2.pbs script
   i) Make sure your job emails you when is starts, ends and aborts
   ii) Make a guess and for enough RAM to run the program
b) Submit your edited Job script , look at your running Job with the following commands, look at the ,memory used by your job
   i) "squeue –u $USER"
   ii) "scontrol show job <jobid>"
   iii) Did your job run successfully? Or fail because of a lack of memory?
   iv) If your job failed due to a lack of memory, increase the maximum memory requested and resubmit your job, and go back to point b)
c) Look at the email reporting on your job success, how much resources were reported used. Compare the memory used to the reported memory in point c).
d) Edit job script and request an appropriate amount of memory to run the Job .
e) Submit your new job
f) Verify that the jobs runs successfully.

21) GPUs

a) Submit a job asking for:
   i) 1 gpu
   ii) 1 cpu
   iii) Has a maximum wall time of 10 minutes
   iv) Sleeps 500 seconds
b) Runs the command scontrol

22) Software licenses and generic resources
*** Currently not implemented on the cluster**

a) Submit a job asking for that asks for
   i) 2 cpus
   ii) 2 sas licenses
b) Try to see resources used by your job, use the scontrol command:

23)Full nodes

a) Submit a job asking for that asks for:

   i) 4 tasks
   ii) 1 node
   iii) Not to run on any nodes with other jobs
   Useful if you are trying to debug your job

b) See if you can see which nodes your job is running on.
   i) "scontrol show job <jobid>"

24) Job dependencies

a) Submit a serial job named dep1, that has:
   i) Walltime of 2:00
   ii) Sleeps 120 seconds
   iii) Submit a serial job 22b waits until job dep2 is done
   iv) Walltime of 2:00
   v) Sleeps 120 seconds
b) Look at job dep2 with "scontrol show job <jobid>"
c) Run the command "squeue -u $USER" ls
d) Verify that Job dep1 complete before dep2 starts

25) This question has been removed

26) Job using temporary directory

a) Submit a job that runs in the temporary directory used no more than 1GB of space,

27)Job environment variables.

a) Submit a serial job that prints the partition(s) that the job was ran in

28)Multiple accounting groups (This exercise will only be available to users with RAC allocated groups or multiple accounting groups. The answer will need to be modified with your accounting group.)

a) Submit a Job to a non default accounting group, that asks for 1 proc
b) Try to see which accounting group your job belongs to, use the scontrol command:
   i) "scontrol show job <jobid>"

29)Basic Job info

a) Use the “squeue -u” and “scontrol show jobid -dd <jobid>” commands to find out how many jobs your have running, queued, in hold state or complete.

b) Use the “showq –b” command to see how many jobs are in what state?

30) Examining a job

a) Start a Job

b) Examine its priority with ”sprio”

c) run scontrol show jobid -dd <jobid> and determine how much RAM the Job asks for/used

31)Job holds

a) See if any of your jobs in the queue have any job holds, if so identify the hold and the reason why.

32) Cluster info

a) How many idle nodes are on the cluster “sinfo --states=idle”

b) How many nodes are down and drained “sinfo –R”

33)Show some detailed information