



# Jobs and VNC

# Jobs and VNC

---

- Daniel Trahan, Andrew Monaghan
- Email: [Daniel.Trahan@Colorado.edu](mailto:Daniel.Trahan@Colorado.edu), [Andrew.Monaghan@Colorado.edu](mailto:Andrew.Monaghan@Colorado.edu)
- RC Homepage: <https://www.colorado.edu/rc>

Sign in! <http://tinyurl.com/curc-names>

- Slides available for download at:  
[https://github.com/ResearchComputing/CHANGE\\_2019](https://github.com/ResearchComputing/CHANGE_2019)

*Adapted from slides by Andrew Monaghan, Aaron Holt, John Blaas, and Shelley Knuth: [1](#), [2](#), [3](#), [4](#).*



# Outline

---

- Jobs Refresher
- Batch Jobs
- Interactive Jobs



# Jobs and Job Scripting

---

- Job – a system allotment of resources that run a particular application.
- Slurm resource manager
- Because Summit and Blanca are shared resource amongst a variety of groups on campus, Users must run their applications through jobs.
  - Ensures everyone can utilize the system
  - No one person taking up too much of the system!
- Blanca is a bit different, ICS owns a portion of the system that is solely shared between ICS researchers.
  - Substantially shorter wait times!



# Why Jobs on VNC?

---

- When accessing Blanca via VNC, you are still accessing the cluster via login nodes.
- Can utilize some basic visualization but heavy computation should still be reserved for the actual cluster.
- How can we submit jobs from our VNC session?
  - Terminal!



# Job Types

---

- Batch Jobs
  - Non-interactive allotment of resources that run applications in the background
  - Think “baking a batch of cookies”
  - Useful for:
    - Applications that take a substantial amount of time
    - Non-interactive Applications
- Interactive Jobs
  - Interactive allotments or resources where user run their applications manually
  - Useful for:
    - Debugging Applications
    - Running GUI Applications



# Submitting Batch Jobs

---

- **sbatch** command: submit a batch job to Slurm
- The sbatch command usually takes in 1 parameter: A job script.
  - Job scripts provide all information on what is needed for the job.
  - Parameters can be overwritten or added externally by specifying the parameter as a flag.
- Example:

```
sbatch test.sh
```

or

```
sbatch test.sh --time=02:00:00
```



# Job Parameters

---

- Allocation: `--account=<account-name>`
- Partition: `--partition=<partition-name>`
- Number of nodes: `--nodes=<nodes>`
- Number of Tasks: `--ntasks=<number-of-tasks>`
- Quality of service: `--qos=<qos>`
- Reservation: `--reservation=<name>`
- Wall time: `--time=<wall-time>`
- Job Name: `--job-name=<jobname>`
- Output File: `--output=<outputname>`

*More on slurm commands:  
<https://slurm.schedmd.com/quickstart.html>*

*FYI: You do NOT actually type <> above – this designates something specific you as a user must enter about your job*



# ICS QoS, Account, and Partition

---

- When using the ICS Blanca nodes, job scripts must specify the following information:

```
--account=blanca-ics  
--partition=blanca-ics  
--qos=blanca-ics
```

- Users of Blanca also have access to a low-priority preemptible QoS that will attempt to run your job on any available node on Blanca
  - Preemptable jobs will be booted off the node and restarted at a later time if a high priority user runs a job on the node.

```
--qos=preemptable
```



# Submitting Your First Job!

---

Submit a slurm job with the following instructions:

1. The job should run the Linux “hostname” command
2. The job script will be named: `submit_hostname.sh`
3. The job will run on 1 core of 1 node
4. We will request a 1 minute wall time
5. Run from the blanca-ics QOS
6. Run on the blanca-ics partition
7. Use the ‘change’ reservation
  - *This is only for this workshop*



# submit\_hostname.sh

---

```
#!/bin/bash

#SBATCH --nodes=1                      # Number of requested nodes
#SBATCH --ntasks=1                      # Number of requested cores
#SBATCH --time=0:01:00                    # Max wall time
#SBATCH --qos=blanca-ics                # Specify QOS
#SBATCH --partition=blanca-ics          # Specify Summit Haswell nodes
#SBATCH --output=hostname_%j.out        # Rename standard output file
#SBATCH --job-name=hostname             # Job submission name
#SBATCH --reservation=change           # Reservation (workshop only)

# Written by:      Shelley Knuth, 15 July 2016
# Updated by:     Andy Monaghan, 8 March 2018
# Purpose:        Demonstrate how to run a batch job on RC resources

# purge all existing modules
module purge

hostname
```

# Running the job script

---

Submit the job:

```
$ sbatch submit_hostname.sh
```

Look at the job output:

```
$ cat hostname_<job-id>.out
```

Check the status of the job:

(\*note that *<job-id>* is your job number)

```
$ squeue / $ squeue -u <user> /
```

```
$ squeue -q <qos>
```

...or

```
$ sacct / $ sacct --format=<options>
```

...or

```
$ scontrol show job <job number>
```

[More on slurm commands: \*https://slurm.schedmd.com/quickstart.html\*](https://slurm.schedmd.com/quickstart.html)

---



# Your turn

---

- Create a Slurm job script and submit it as a job, with the following instructions:
  1. Name it 'submit\_sleep.sh'
  2. The job should run first the whoami command, then the Linux “sleep” command for 30 seconds, then the hostname command
  - Syntax for these Linux commands is:

```
echo "I am" `whoami`  
echo "Running on host" `hostname`  
echo "Starting Sleep"  
sleep 30  
echo "Ending Sleep. Exiting Job!"
```

# Submit\_sleep.sh

---

```
#!/bin/bash

#SBATCH --nodes=1          # Number of requested nodes
#SBATCH --ntasks=1          # Number of requested tasks
#SBATCH --time=00:01:00      # Max wall time
#SBATCH --qos=blanca-ics    # Specify QOS
#SBATCH --partition=blanca-ics # Specify Summit Haswell nodes
#SBATCH --output=sleep_%j.out # Rename standard output file
#SBATCH --job-name=sleep     # Job submission name
#SBATCH --reservation=change # Reservation (workshop only)

# purge all existing modules
module purge

echo "I am" `whoami`
echo "Running on host" `hostname`
echo "Starting Sleep"
sleep 30
echo "Ending Sleep. Exiting Job!"
```

# Interactive jobs

---

- Batch jobs are great for running applications in the background
- Sometimes we want to work on program in real time
  - Great for testing, debugging, visualization
- For example, let's run an interactive Python job...



# Running an interactive job

---

- To work with R interactively, we request time from Summit
- When the resources become available the job starts
- Commands to run:

```
sinteractive --time=00:10:00 --qos=blanca-ics
```
- Once we receive a prompt, then:

```
module load python
python
```
- Once we finish we must exit! (job will time out eventually)



# Notes on Interactive Jobs

---

- Interactive Jobs continue to run if a terminal is closed while the terminal is shelled into the Job's allotment of resources
  - You must run the exit command when you are done!
- Quirks about the interactive jobs shell:
  - Cannot scroll up to see previous information
  - Graphical bugs are common
- By default Interactive Job run with 1 core on 1 node.
  - You can specify more or less resources with the flags:

```
sinteractive --nodes=2 --ntasks=24 --time=01:00:00 --qos=blanca-ics
```

- *Any resource that can be specified with sbatch can be used with sinteractive!*



# Thank you!

---

- Please fill out the survey: <http://tinyurl.com/curc-survey18>
- Sign in! <http://tinyurl.com/curc-names>
- Contact information:  
[rc-help@Colorado.edu](mailto:rc-help@Colorado.edu)  
[Daniel.Trahan@Colorado.edu](mailto:Daniel.Trahan@Colorado.edu)
- Slides and Examples from this course:  
[https://github.com/ResearchComputing/CHANGE\\_2019](https://github.com/ResearchComputing/CHANGE_2019)

