MARCC Tutorial

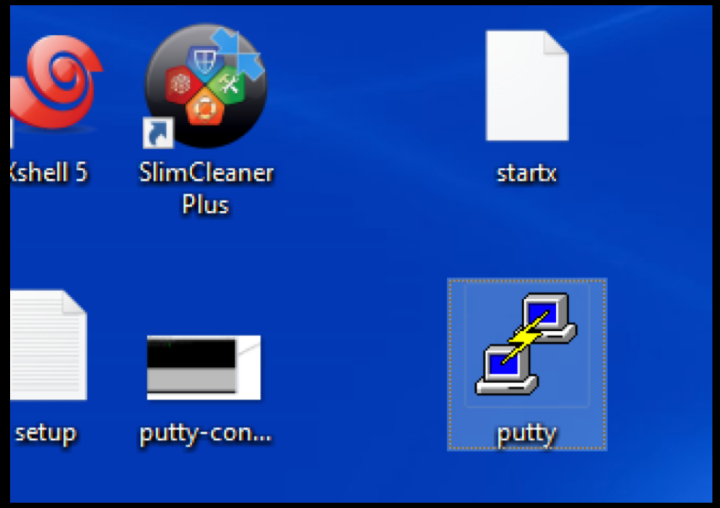
Introduction to BASH, LMOD, SLURM

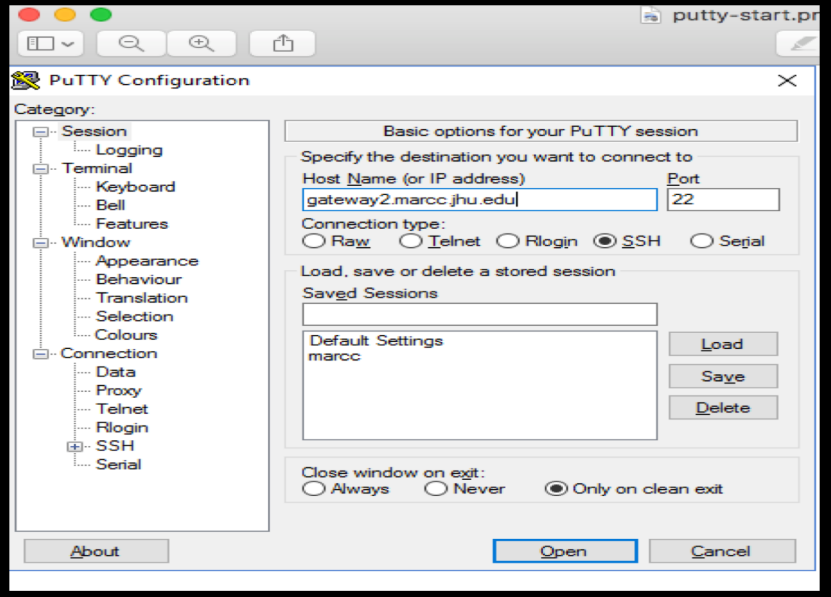
Shells

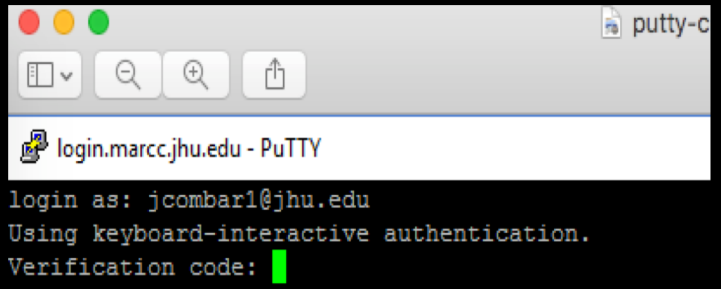
- BASH: Bourne Again SHell (Recommended)
- DASH
- SH -> /bin/bash
- ZSH
- TCSH

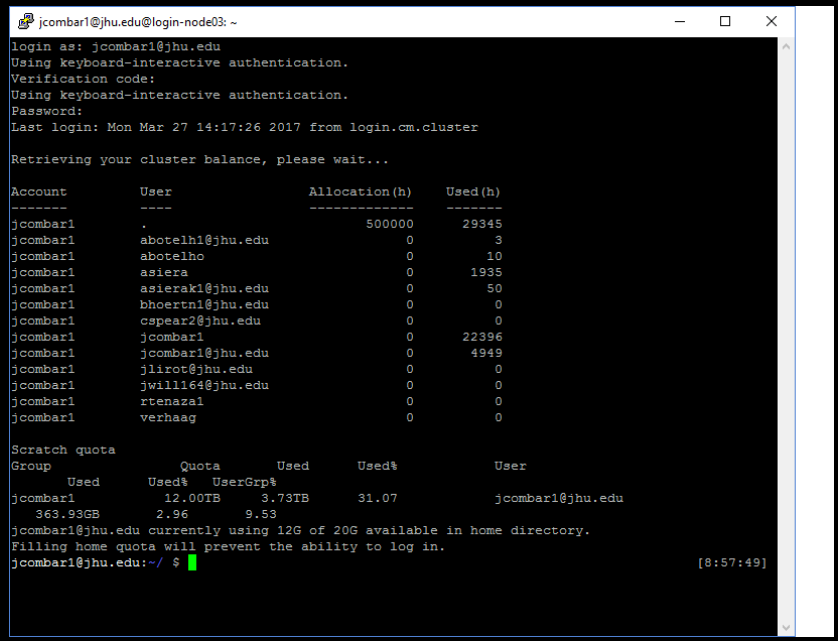
Connecting

- ssh gateway2.marcc.jhu.edu -l jhedid@jhu.edu
- scp FILE userid@umd.edu@dtn2.marcc.jhu.edu:~/scratch/.
- scp -r DIR userid@umd.edu@dtn2.marcc.jhu.edu:~/scratch/.
- WINDOWS: Using Putty: http://www.putty.org


Xshell 5
SlimCleaner Plus
startx
setup
putty-con...
putty


putty-start.pr
PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
gateway2.marcc.jhu.edu
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
marcc
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel


putty-c
login.marcc.jhu.edu - PuTTY
login as: jcombar1@jhu.edu
Using keyboard-interactive authentication.
Verification code:


jcombar1@jhu.edu@login-node03: ~
login as: jcombar1@jhu.edu
Using keyboard-interactive authentication.
Verification code:
Using keyboard-interactive authentication.
Password:
Last login: Mon Mar 27 14:17:26 2017 from login.cm.cluster
Retrieving your cluster balance, please wait...
Account User Allocation(h) Used(h)
jcombar1 . 500000 29345
jcombar1 abotelh1@jhu.edu 0 3
jcombar1 abotelho 0 10
jcombar1 asiera 0 1935
jcombar1 asierak1@jhu.edu 0 50
jcombar1 bhoertn1@jhu.edu 0 0
jcombar1 cspear2@jhu.edu 0 0
jcombar1 jcombar1 0 22396
jcombar1 jcombar1@jhu.edu 0 4949
jcombar1 jlirot@jhu.edu 0 0
jcombar1 jwill164@jhu.edu 0 0
jcombar1 rtenaza1 0 0
jcombar1 verhaag 0 0
Scratch quota
Group Quota Used Used% User
Used Used% UserGrp%
jcombar1 12.00TB 3.73TB 31.07 jcombar1@jhu.edu
363.93GB 2.96 9.53
jcombar1@jhu.edu currently using 12G of 20G available in home directory.
Filling home quota will prevent the ability to log in.
jcombar1@jhu.edu:~/ $
[8:57:49]

Navigating

- cd .. #### Change to parent dir
- cd - #### Change dir back to previous
- cd (cd ~) #### change dir HOME
- ls -a -l -t -r #### list files in cwd
- pwd #### print current work dir
- du -kh - -max-depth=1

Home

- ~
- ~/scratch (Soft link)
- ~/work
- ~/data

Environment variables

- printenv | grep -i path
- echo : built-in command—-> writes arguments to standard output
- echo $PATH
- echo $LD_LIBRARY_PATH
- echo $MANPATH
- echo $PYTHONPATH

Finding files

- Directories (folders) are just special files
- find . -type d
- locate
- which
- type

Finding commands

- Commands are *mostly* just executable files
- which python
- which ml <— won’t work because ml is a function
- type ml
- type python

Completion

- Line commands

- TAB is your friend
- ls /software/apps/matlab/R201[TAB]

History

- history
- !1000
- !-2
- !!
- up, down
- CTRL-r, CTRL-a

Regex

- www.regexr.com
- *
- ` ` ##### ls `which python`
- | (pipe) ##### ls -l | more
- | (or) ‘ #### egrep ‘and|or|to’ README
- / \

Manipulating files

- mkdir ### Create new dir
- rm -i -r -f ### be careful DELETE files
- cp -r
- ln -s
- alias ### alias rm =“rm -i”

Text files

- nano
- vim, emacs
- less, more
- cat, sed, grep, cut, join, paste
- head, tail

Streams

- stdin, stdout, stderr
- >, >>, |, &
- tee

Configuring BASH

- Hidden files names start with with a period
- ~/.bashrc
- ~/.bash_profile
- ~/.bash_history
- ~/.bash_logout

File permissions

- (u=user, g=group, o=others)
- r=read, w=write, x=execute
- chmod g+r <file>
- chmod -w <file>
- If you own the file, you can never remove your ability to delete it or change its permission

LMOD: searching

- module list
- module avail
- module spider python
- module keyword python
- module show python/2.7.10
- module help python/2.7.10

LMOD: loading

- module load python/3.4.2
- module swap python/3.4.2 python/2.7.9
- module unload python/2.7.9

LMOD save/load

- module save mymods
- cd ~/.lmod.d
- cat python
- module restore mymods

LMOD: use.own

- mkdir pwgen; cd pwgen
- wget http://dl.fedoraproject.org/pub/epel/6/x86_64/pwgen-2.07-1.el6.x86_64.rpm
- rpm2cpio pwgen-2.07-1.el6.x86_64.rpm | cpio -idmv
- mv usr 2.07-1

LMOD: use.own

- cd ~/privatemodules
- mkdir pwgen
- nano 2.07-1.lua

pwgen/2.07-1.lua

```
local help_message = [[

pwgen generates random strings for use as passwords

]]


help(help_message,"\n")


whatis("Name: pwgen")

whatis("Version: 2.07-1")


prepend_path("PATH","/home/abotelho/apps/pwgen/2.07-1/bin")

prepend_path("MANPATH","/home/abotelho/apps/pwgen/2.07-1/share/man")
```

LMOD: use.own

- module load use.own
- module load pwgen
- module list
- man pwgen
- pwgen

Scripts

#!/bin/bash

#!/bin/dash

#!/bin/sh

#!/usr/bin/python <— make sure you have it right!

SLURM Scripts

```
#SBATCH --partition=debug
#SBATCH --time=d-hr:min:sec
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=2
#SBATCH --mail-type=end
#SBATCH --mail-user=user@jhu.edu

module load use.own
module load pwgen
pwgen
```

SLURM Scripts

— Using GNU Parallel to run a batch of jobs on one node

```
#SBATCH --partition=debug
#SBATCH —time=2:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mail-type=end
#SBATCH --mail-user=user@jhu.edu

module load parallel gaussian
#set gaussian environment
mkdir -p /scratch/users/$USER/$SLURM_JOBID
export GAUSS_SCRDIR=/scratch/users/$USER/$SLURM_JOBID

cat my-list | parallel -j 20 —joblog LOGS “g09 {}”

### ls *.com > my-list
### -j 20      run only 20jobs at a time
###  - -joblog  LOGS  add all information about the jobs into LOGS
```

SLURM Scripts

— Using Job Arrays, limit 72 running jobs out of 500
#SBATCH --partition=shared
#SBATCH —time=2:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mail-type=end
#SBATCH --mail-user=user@jhu.edu
#SBATCH —array=1-500%72

module load parallel gaussian
#set gaussian environment
mkdir -p /scratch/users/$USER/$SLURM_JOBID
export GAUSS_SCRDIR=/scratch/users/$USER/$SLURM_JOBID

file=$(ls *.com | sed -n ${SLURM_ARRAY_TASK_ID}p)
echo $file
g09 $file

list all files in directory into a variable $file
Run 72 jobs at a time until done
This script will submit all 500 jobs but will group them so only 72 will run at a time

SLURM Scripts

```
#!/bin/bash -l
#SBATCH --job-name=largearray
#SBATCH --partition=shared
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=02:10:00
#SBATCH --output=array_%A-%a.out
#SBATCH --array=1-20
#Set the number of runs that each SLURM task should do
PER_TASK=25
# Calculate the starting and ending values for this task based on the SLURM task and the number of runs per task.
START_NUM=$(( ($SLURM_ARRAY_TASK_ID - 1) * $PER_TASK + 1 ))
END_NUM=$(( $SLURM_ARRAY_TASK_ID * $PER_TASK ))
# Print the task and run range
echo This is task $SLURM_ARRAY_TASK_ID, which will do runs $START_NUM to $END_NUM
# Run the loop of runs for this task.
for (( run=$START_NUM; run<=END_NUM; run++ )); do
  echo This is SLURM task $SLURM_ARRAY_TASK_ID, run number $run
module load gaussian
mkdir -p /scratch/users/$USER/$SLURM_JOBID
export GAUSS_SCRDIR=/scratch/users/$USER/$SLURM_JOBID

file=$(ls *.com  | sed -n ${SLURM_ARRAY_TASK_ID}p)
echo $file
g09 water$run

done
```

Submitting jobs

- sbatch pwgen.scr (qsub)
- squeue -u $USER (qstat -a)
- sqme
- sacct
- scontrol show job job-id
- scancel (qdel)
- sinfo -p shared

Interactive jobs

- interact -usage
- interact -p debug -t 60