Triton Shared Compute Cluster (TSCC) Quick Start

Burak Himmetoglu
Supercomputing Consultant

Enterprise Technology Services &
Center for Scientific Computing
University of California
Santa Barbara

e-mail: bhimmetoglu@ucsb.edu

General Information

Performance: 80 Tflops
General computing nodes: Duel socket 8-core 2.6 GHz Intel Xeon E5-2670
GPU nodes: Host processors: Duel-socket 6-core Intel Xeon E5-2630,
GPU: 4 NVIDIA GeForce GTX 980

TSCC user guide: detailed information on the cluster and installed software:

http://www.sdsc.edu/support/user_guides/tscc.html

http://www.sdsc.edu/support/user_guides/tscc-quick-start.html

Access

- For access, send the public key (id_rsa.pub) to: bhimmetoglu@ucsb.edu
- Your ssh key will allow passwordless connection to Triton.
- You can connect only using the computer where the key is generated.

How to generate SSH keys:

Linux:

ssh-keygen

Windows (using Putty):

http://kb.site5.com/shell-access-ssh/how-to-generate-ssh-keys-and-connect-to-your-account-with-putty/

http://wiki.joyent.com/wiki/display/jpc2/Manually+Generating+Your+SSH+Key+in+Windows

MAC:

http://wiki.joyent.com/wiki/display/jpc2/Manually+Generating+your+SSH+Key+in+Mac+OS+X

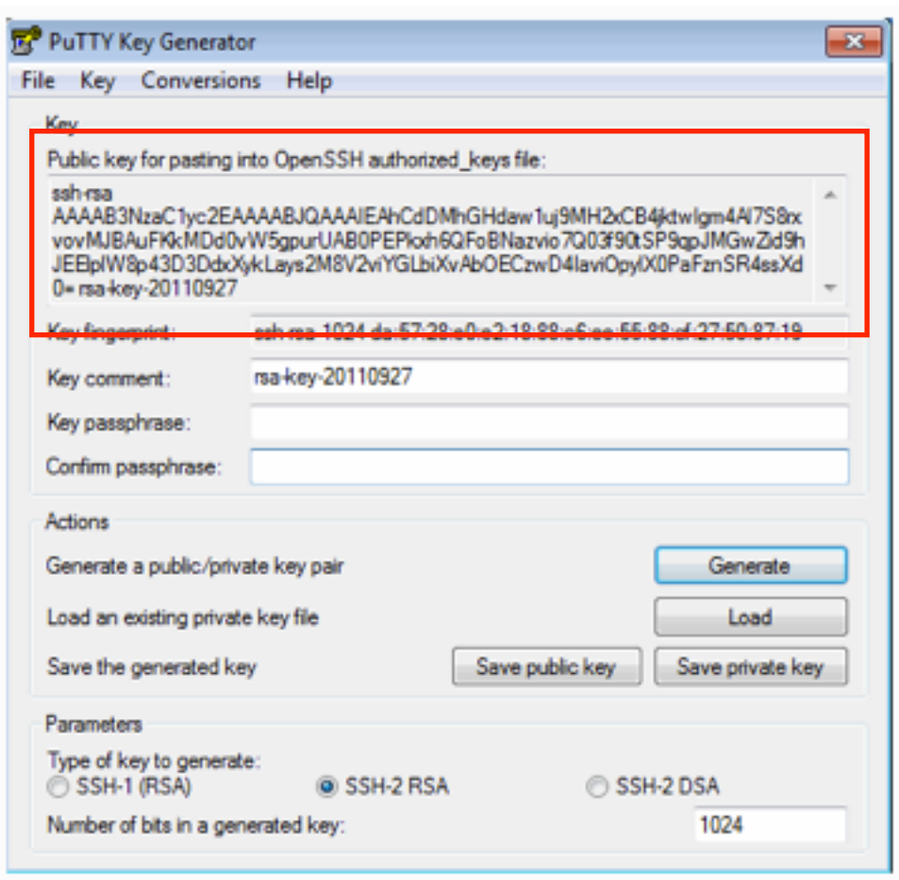
Logging in:

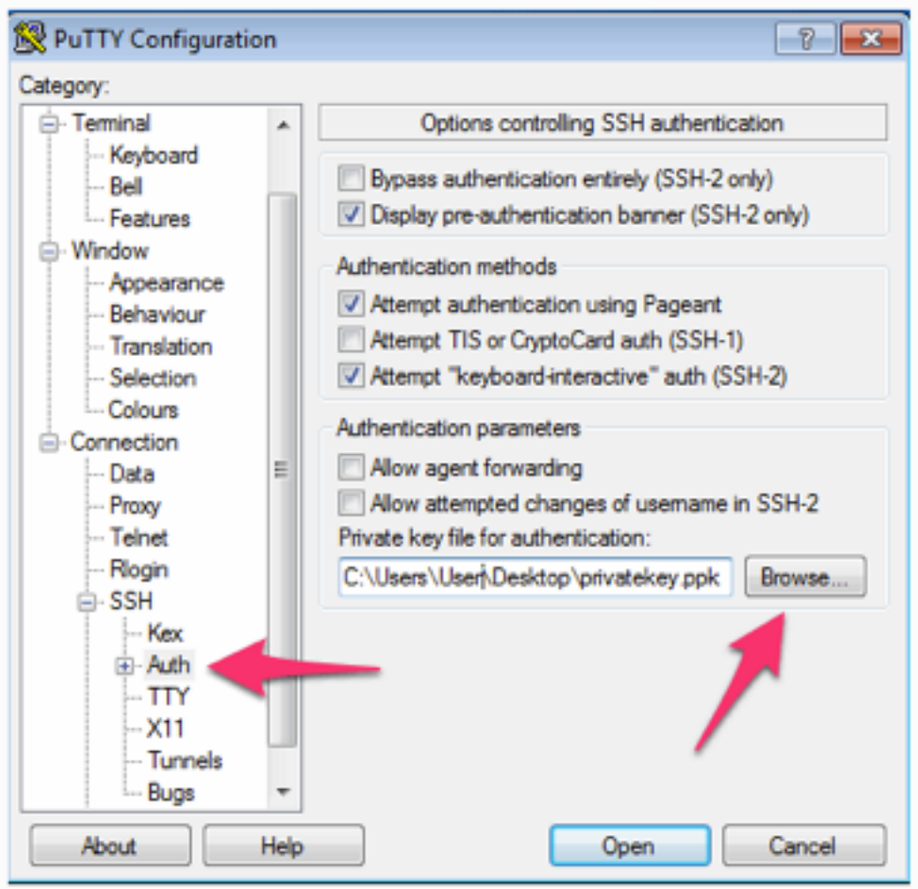
ssh username@tscc-login.sdsc.edu

Access

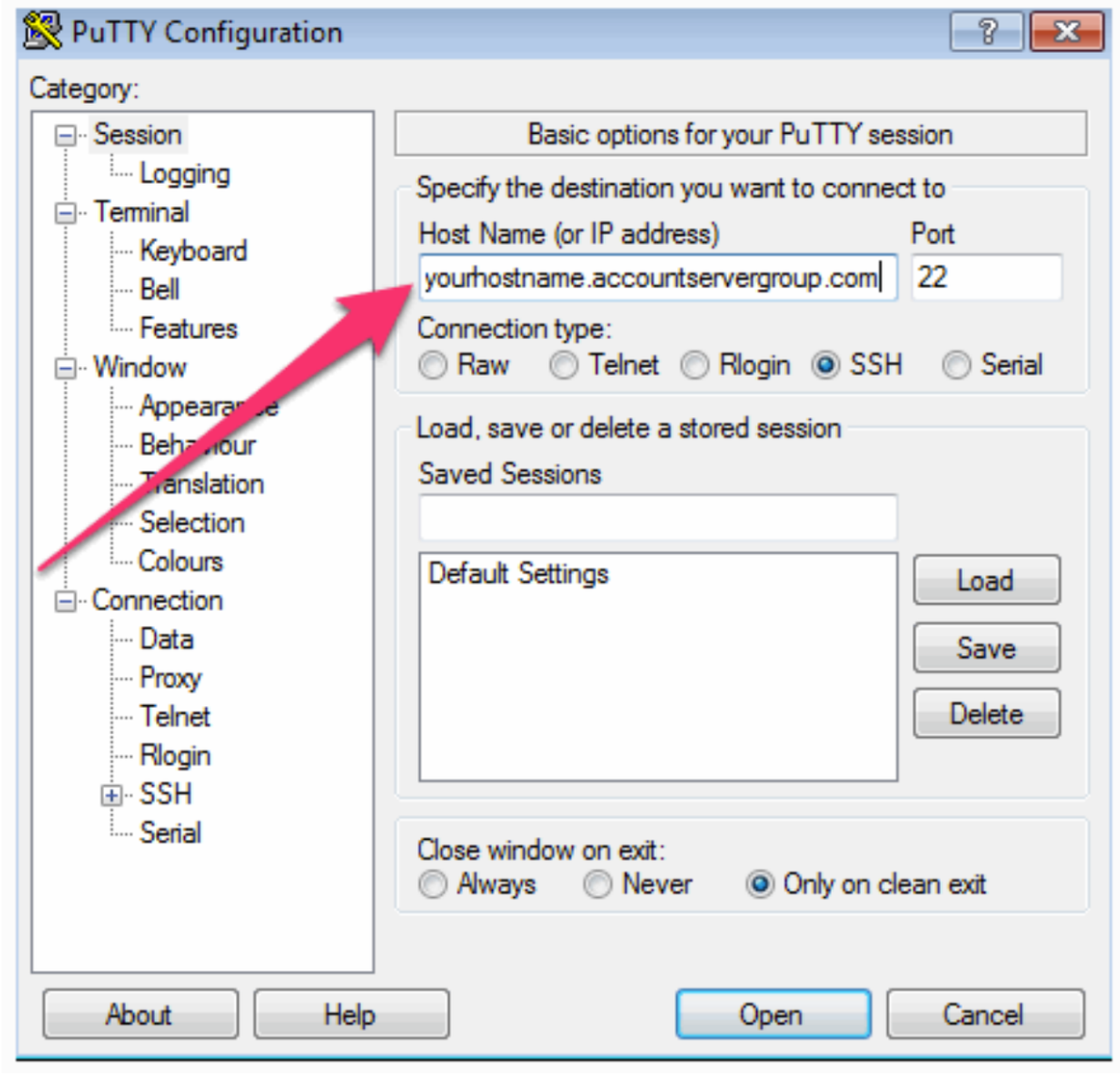
- No need to create a “passphrase”, just press enter leaving it blank when you are asked
- Your public key is included in your home directory: `.ssh/id_rsa.pub`
- Please send the public key as a single line.

Using Putty





Access



yourhostname: tscc-login.sdsc.edu

Access

Check balance:

```
gbalance -u username
```

- In the beginning of the class, each student is assigned 500 hrs
- If you run out of hours please e-mail bhimmetoglu@ucsb.edu asking for a supplement
- The request goes to SDSC, therefore please send supplement requests before Fridays

File transfer and X windows

Open a terminal window

To copy the file “file.txt" from local computer to TRITON:

```
scp file.txt username@tscc-login.sdsc.edu:~/folder
```

To copy the file file.txt from TRITON to local computer:

```
scp username@tscc-login.sdsc.edu:~/folder/file.txt  ./
```

X Window, for graphics (e.g plotting)

```
ssh -X username@tscc-login.sdsc.edu
```

In Mac, you need XQuartz installed.

File transfer and X windows

X Window System Server for Windows:

http://sourceforge.net/projects/xming/

Xming is the leading X Window System Server for Microsoft Windows 8/7/Vista/XP. It is fully featured, small and fast, simple to install and because it is standalone native Microsoft Windows, easily made portable (not needing a machine-specific installation).

WinSCP:

http://winscp.net/eng/index.php

WinSCP is an open source free SFTP client, FTP client, WebDAV client and SCP client for Windows. Its main function is file transfer between a local and a remote computer. Beyond this, WinSCP offers scripting and basic file manager functionality.

Running Jobs

Caution: Do not run jobs on the login node!
Submit jobs to the compute notes, using scripts.

Example: Job that uses MPI parallelization

ex_mpi.pbs

```
#!/bin/bash
#PBS -q hotel
#PBS -N hello
#PBS -l nodes=1:ppn=4
#PBS -l walltime=0:05:00
#PBS -o hello-out
#PBS -e hello-err
#PBS -M your_email
#PBS -m mail_option

cd /home/work

mpirun -v -machinefile $PBS_NODEFILE -np 4 mpi_hello > h-out
```

mail_option: n (no e-mail sent), b (e-mail sent when job begins execution), e (when job terminates)

Running Jobs

Submission: `qsub ex_mpi.pbs`

Check status of your job(s): `showq -u your_username`

Cancel a running job: `qdel job_id`

Some Notes:

- If your code does not use MPI parallelization, do not ask for more than 1 node
- For example, if you use only OpenMP, use only nodes=1
- Jobs that require more resources will wait longer in the queue
- Try to minimize walltimes, by estimating the time that your code will run.
- Calculate the resources that will be spent:

```
#PBS -l walltime=01:00:00
#PBS -l nodes=2:ppn=8
```

⟶ (nodes x procs) x wtime = 2 x 8 x 1 = 16 hrs

Running Jobs (GPU)

First compile your code:

```
$ module load cuda
$ nvcc hello_cuda.cu -o hello_cuda.x
```

ex_cuda.pbs

```
#!/bin/bash
#PBS -q gpu-hotel
#PBS -N hello
#PBS -l nodes=1:ppn=1
#PBS -l walltime=0:05:00
#PBS -o hello-out
#PBS -e hello-err
#PBS -M your_email
#PBS -m mail_option

cd /home/work

./hello_cuda.x > out
```

```
$ qsub ex_cuda.pbs
```