LINKSCEEM

Linking Scientific Computing in Europe and the Eastern Mediterranean

HPC Roadshow

Hands on training session for core skills

Overview

- Obtaining an account
- How to Generate a Public/Private Key Pair
- How to Access the System (ssh)
- Software Environment – Modules
- How to Find Cluster Status and Job Summary
- Compiling and Submitting Jobs - Examples

Obtaining an Account

- **To gain access to one of the LinkSCEEM HPC resources after your application has been accepted:**
  - Print, complete, sign and fax an Acceptable Usage Policy (AUP) for the resource you have been granted access to
    - Can be found in the following link
    - As “Project Reference code” put the project code (eg. lspre100s1) corresponding to your linklings submission. For the educational access projects as "Projects Reference code" type "Educational Access".
  - Request a user account on the allocated HPC resource(s) by completing the Access Issues form
  - You will then be contacted by the user support team to provide your public key

How to generate a Public/Private key pair

Mac OS X/Linux/Unix Keypairs

- **Create a public and private key pair:**

```
ssh-keygen -t rsa
```

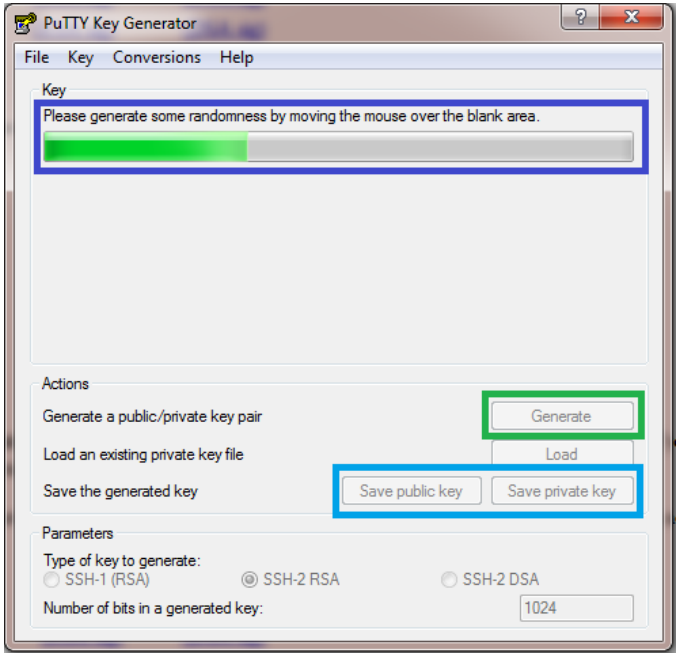
- **You will prompted for a filename for saving the private key. You should press Enter to use the default name**

```
/home/username/.ssh/id_rsa
```

- **You will prompted for a password to protect your private key which you will also confirm**
  - ALWAYS use such a password to protect your private key
- **ssh-keygen will then create a private and public key**
  - The public key will have a .pub extension (e.g., id_rsa.pub)

How to generate a Public/Private key pair

- **Windows**
  - Using Putty
- **You will need to have PuTTy and PuTTygen installed**
  - Download here
- **Run PuTTygen**
  - Click on Generate
  - Run your mouse over the blank area to generate some randomness
  - Save your key pair when done



Important Private/Public Key Notes

- **Always attach a password to your ssh keys**
  - Please note that this is the password which allows access to use the keys
  - To access the HPC resource from any machine you need access to your keys
    - Best to have these on your usb drive
  - ssh –I /media/usbName/keyName (path if in a folder)
    - Will enable you to select the private key you want to use
    - You can thus bypass the default key on the machine you are using
    - By selecting the private key stored on your usb you can access HPC resources from any machine available to you
  - With Putty you just select the key using Browse

How to access the system

- **To access the system you must have access to your private key**
- **With Mac OS X/Linux/Unix:**
  - This should be found in your .ssh folder in your home directory

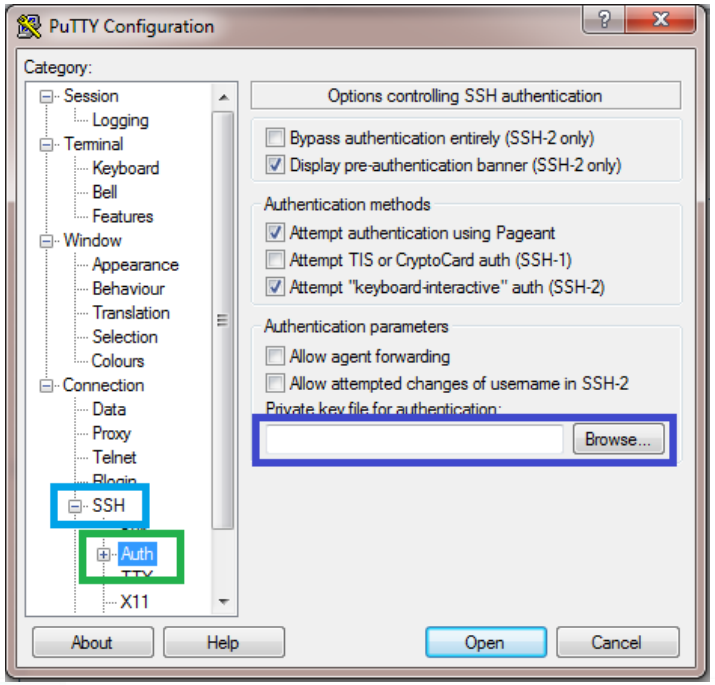
```
ssh username@euclid.cyi.ac.cy
```

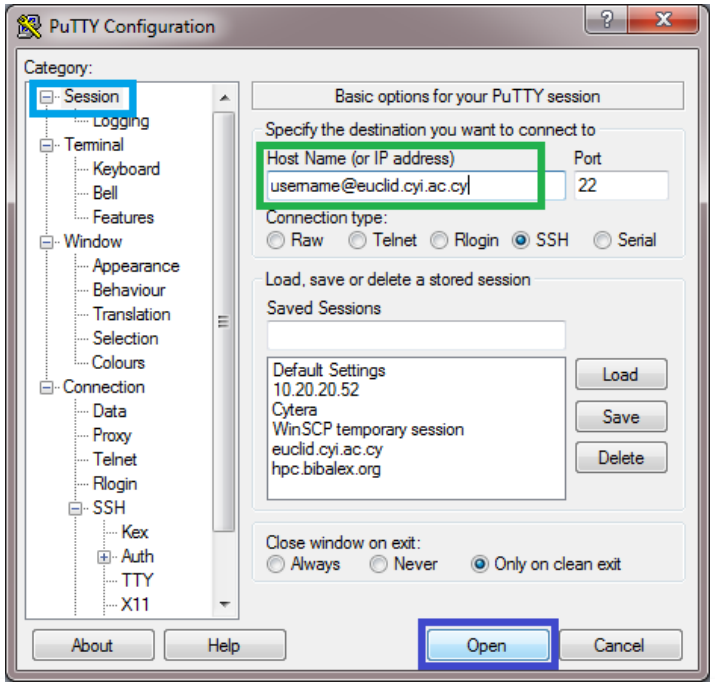
  - Should you encounter a problem with your folder permissions, do the following:

```
chmod 600 .ssh/id_rsa
```

How to access the system

- **With Windows**
  - Use PuTTy
    - First browse for and select your private key file
    - Enter ssh login information





Environment Setup with Modules

- The software environment used on LinkSCEEM systems can be managed via Modules
- Modules facilitate the task of updating applications and provide a user-controllable mechanism for accessing software revisions and controlling combination of versions

```
1 module avail   # lists available modules
2 module list    # lists currently loaded modules
3 module help    # help on a specific module
4 module whatis # brief description of a specific module
5 module display # displays changes by a given module
6 module load X # load a specific module
7 module unload # unloads a specific module
8 module clear   # unloads all modules
```

Common modules and libraries

```
[thekla@euclid ~]$ module avail

------------------------ /opt/software/Modules/versions ------------------------
3.2.9

------------------- /opt/software/Modules/3.2.9/modulefiles --------------------
dot     modules null    use.own

----------------- /opt/software/Modules/modulefiles/compilers ------------------
intel/12.1.1.256 pgi/12.8

-------------------- /opt/software/Modules/modulefiles/mpi ---------------------
mvapich2/1.8-gcc(default)  openmpi/1.4.5-gcc(default)
mvapich2/1.8-intel         openmpi/1.4.5-intel

-------------------- /opt/software/Modules/modulefiles/libs --------------------
cuda/4.2

-------------------- /opt/software/Modules/modulefiles/sys ---------------------
maui/3.3.1   torque/3.0.3
```

Hands on Exercise 1

C code Example

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char ** argv) {
int size,rank;
int length;
char name[80];

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
MPI_Comm_size(MPI_COMM_WORLD,&size);
MPI_Get_processor_name(name,&length);

printf("Hello MPI World! Process %d out of %d on %s\n",rank,size,name);
MPI_Finalize();
return 0;
}
```

Hands on Exercise 1

Batch Script Example

- Script which can be used to define all commands required to submit a job to a HPC resource

```
#!/bin/bash

#PBS -N mpi-hello_world
#PBS -j oe
#PBS -q batch          #PBS –q cpuq (for Cy-Tera CPU queue)
#PBS -l nodes=1:ppn=8

cd $PBS_O_WORKDIR

module load openmpi
mpiexec ./hello
```

Hands on Exercise 1

- **From your training account home directory (cd)**
  - Copy the training example with the following command

```
cp -r /opt/examples/coreskills/ .
```

  - Go the the coreskills directory

```
cd coreskills
```

- **Load the mpi module and compile the code**

```
module load openmpi
mpicc hello.c -o hello
```

Hands on Exercise 1

- **Submit the job**

On Euclid:
```
qsub hello.pbs
```
On Cy-Tera:
```
msub hello.pbs
```

- **Check your job status**

```
qstat
```

- **Check the contents of the output file once your job is completed**

```
cat output_file_name
```

Hands on Exercise 2

- **matrix-matrix multiplication in a number of different scenarios: serial, gpu (using different approaches) and mpi**
  - http://keeneland.gatech.edu/software/sgemm_tutorial
- **From your training account home directory (cd)**
  - Copy the training example with the following command

```
cp -r /opt/examples/roadshow_example/ .
```

- **Go to the rodshow_example directory and load the three modules needed:**

```
cd roadshow_example
module load intel openmpi/1.4.5-intel  cuda
```

Hands on Exercise 2

- **Compile all codes simultaneously using the Makefile**

```
ln -s Makefile.euclid_intel Makefile
make -j 8
```

- **Submit the different scripts, for example:**

```
qsub mm.pbs
qsub mm_cuda.pbs
```

- **Compare the average time needed for the matrix-matrix multiplication from the output files of the different jobs**

Thank you