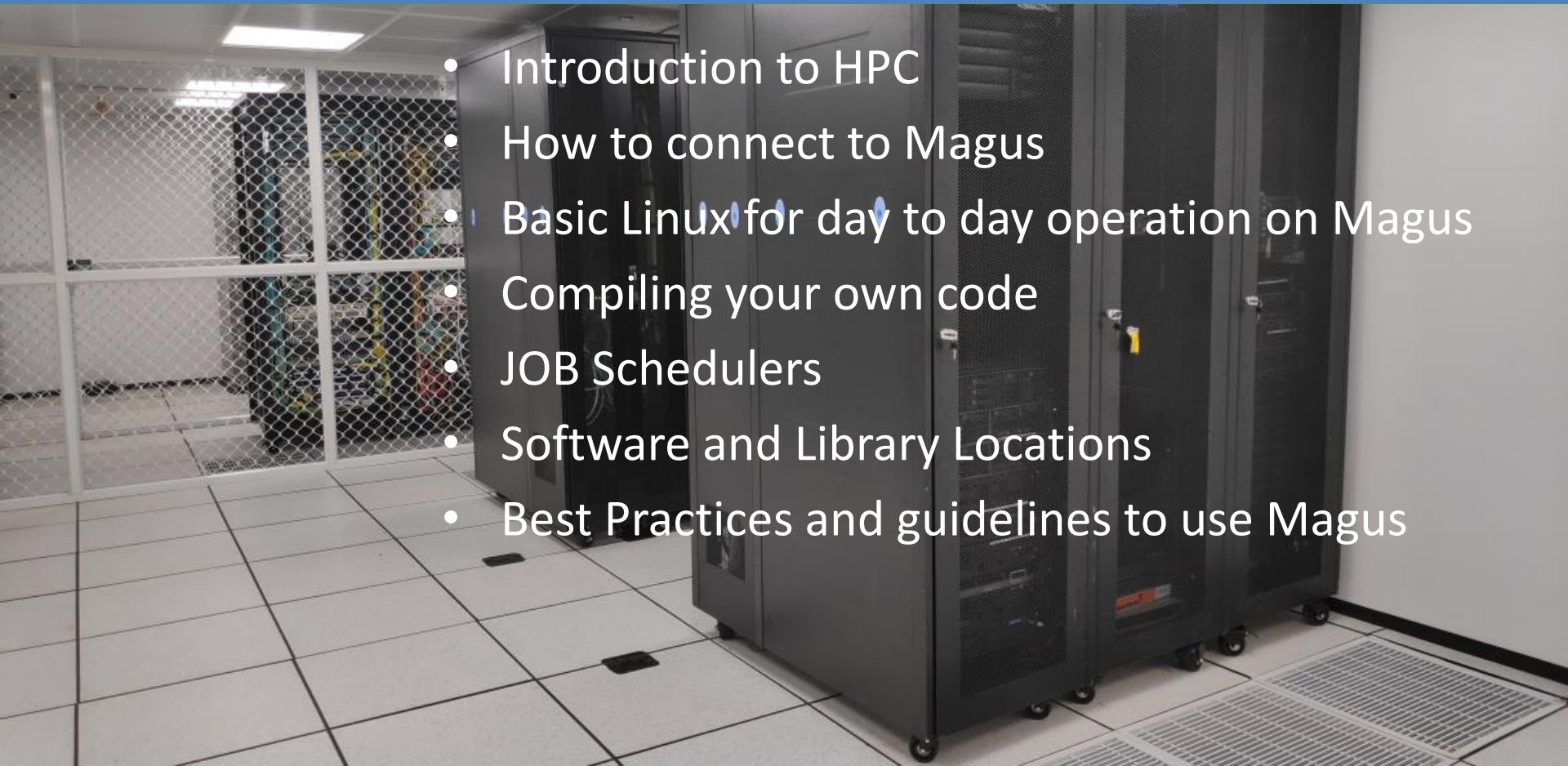

# HPCC- MAGUS- Introduction.

# Objectives





- Introduction to HPC
- How to connect to Magus
- Basic Linux for day to day operation on Magus
- Compiling your own code
- JOB Schedulers
- Software and Library Locations
- Best Practices and guidelines to use Magus

# Introduction



- What is HPC and why is it different to using your desktop?

  *"High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business." -insideHPC*

- Aggregated computing power
- Very large problem sizes
- Multiple problems simultaneously
- Large Data sets

# Parallel Computing



- **Example of Serial Computing.**
  - **(5+2) + (3*4) * 2*(100-(2+4))**
    - *7+(3*4)*2*(100-(2+4))*
    - *7+12*2*(100-(2+4))*
    - *7+12*2*(100-6)*
    - *7+12*2*94*
    - *7+24*94*
    - *7+2256*
    - Result 2263
    - **Total Cycles Used 7**

- **Using Parallel Computing**
  - **Using 3 nodes**

| | Node 1 | Node 2 | Node 3 |
|---|---|---|---|
| Cycle 1 | (5+2) | ( 3*4 ) | ( 2+4) |
| Cycle 2 | 100-6 | 12*2 | Idle |
| Cycle 3 | 24*94 | idle | Idle |
| Cycle 4 | 7+2256 | idle | idle |
| | | | |

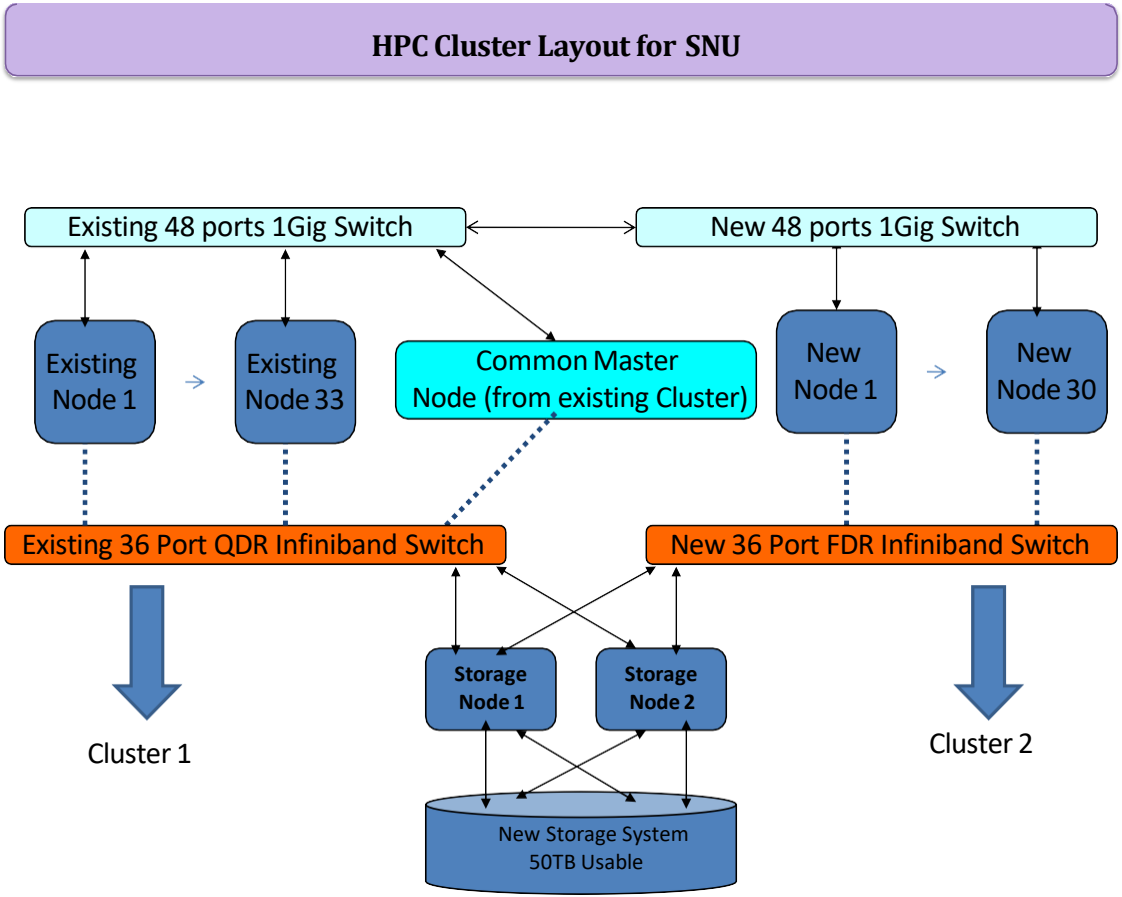
**Result : 2263**

**Total Cycle used 4**

# SNU HPCC- Magus – Key facts

- **Total 62 Compute Nodes.**
- **New Intel Hash well Architecture Processors on 30 Nodes**
- **Total ~ 1000 Cores**
- **Total ~ 6 TB RAM.**
- **IBM GPFS Parallel File System**
- **~30 TF Theoretical Peak Performance Total.**
- **8- High CPU & Memory Nodes**
- **Cluster Management Software :- IBM  Platform HPC**

# SNU HPCC- Magus Architecture

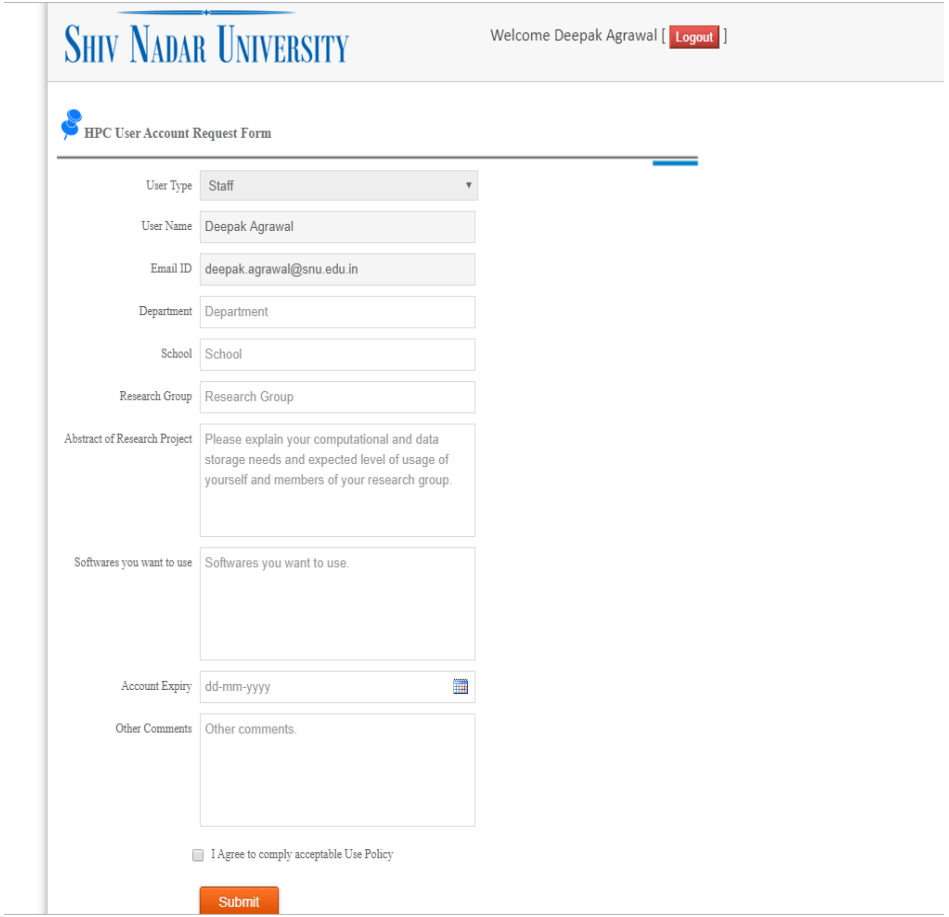


# Magus



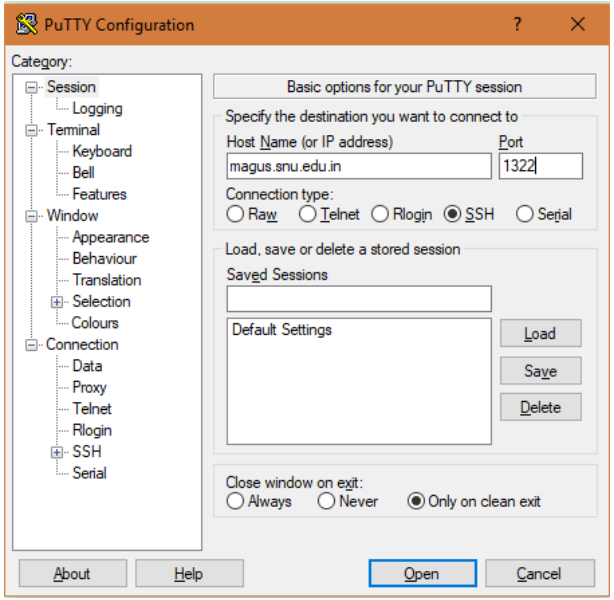
## Request for User Account on HPC.

http://hpc.snu.edu.in/hpcAccount



# How to Connect to Magus

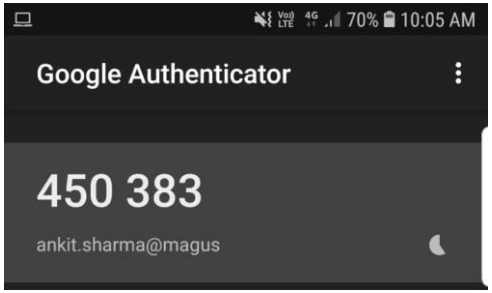
- **SSH**
  - **Host magus.snu.edu.in ( default port 1322 )**
- **From Linux Machine**
  - **$ ssh user@magus.snu.edu.in –p 1322**
- **From Windows Machine using putty**

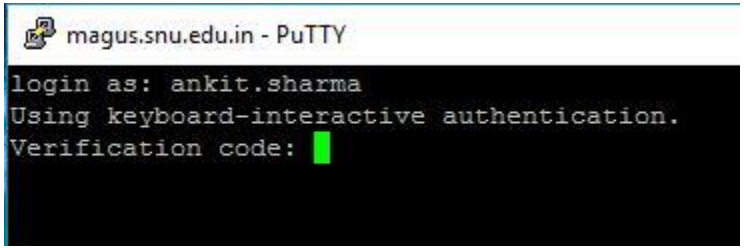


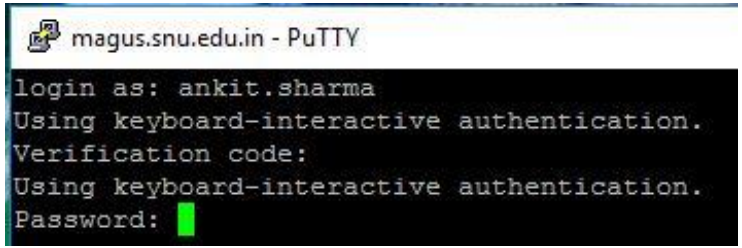
# Connect to Magus from outside SNU Network



- Similar process as you connect within SNU network
- Additional Layer of security with Google Authenticator
- User needs to enter the verification code generated by the Google Authenticator App
- Once successful. Then the user needs to enter the password to login




Reference Url: http://wiki.snu.edu.in/index.php/Google_Authenticator

# How to move data In/Out from HPC



## SCP General Syntax

```
scp source_file_name username@destination_host:destination_folder
```

SCP commonly used flags

-P port : Specifies the port to connect to on the remote host.
-p : Preserves modification times, access times, and modes from the original file.
-q : Quiet mode: disables the progress meter as well as warning
-r : Recursively copy entire directories.
-C : Compress Data while copy.
-v : Verbose mode.

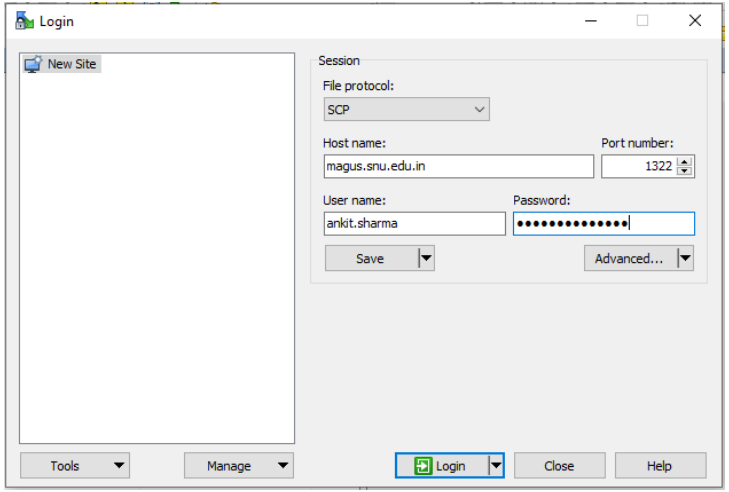
## Sample file transfer

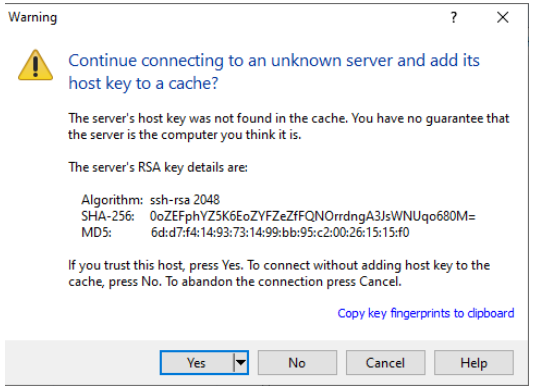
```
[root@backup ~]# scp -rC -P 1322 ankit.txt ankit.sharma@magus.snu.edu.in:/snufs/home/ankit.sharma/
############################################################
#                 Authorized access only!                  #
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
#         All actions Will be monitored and recorded       #
############################################################
Verification code:
Password:
Copyright (C) 2009-2014 Intel Corporation. All rights reserved.
[root@backup ~]#
```

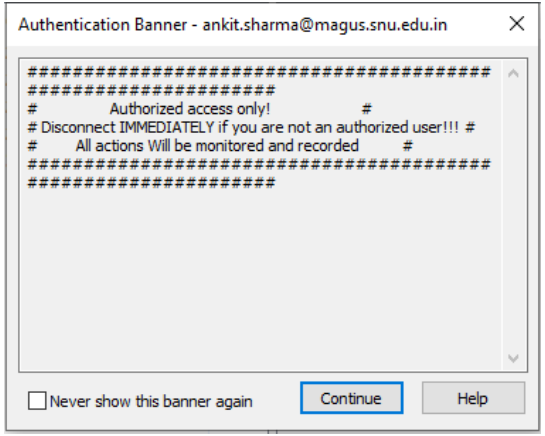
*Best Practise: Data Transfer Location must be your Home Dir: /snufs/home/<user.name>/*

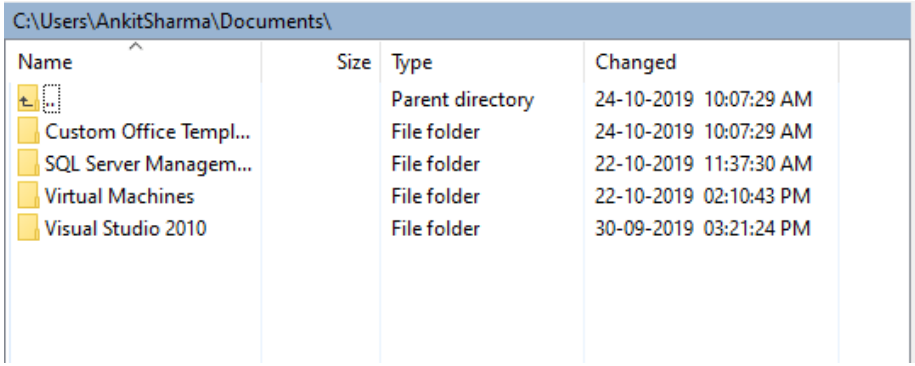
# Data Transfers using WinSCP Windows (Client)

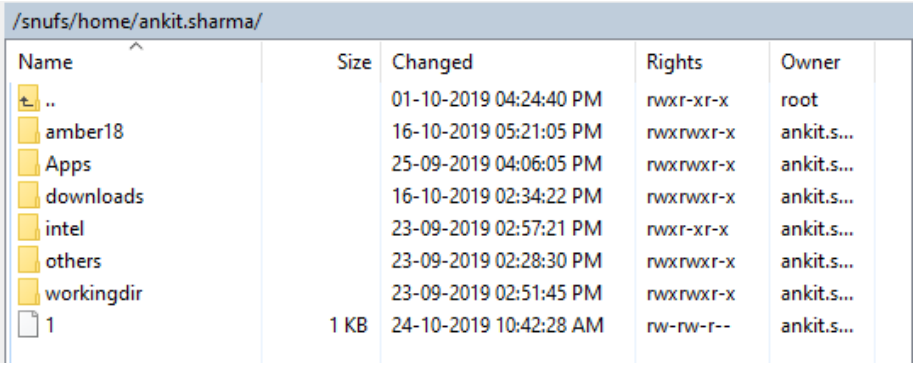




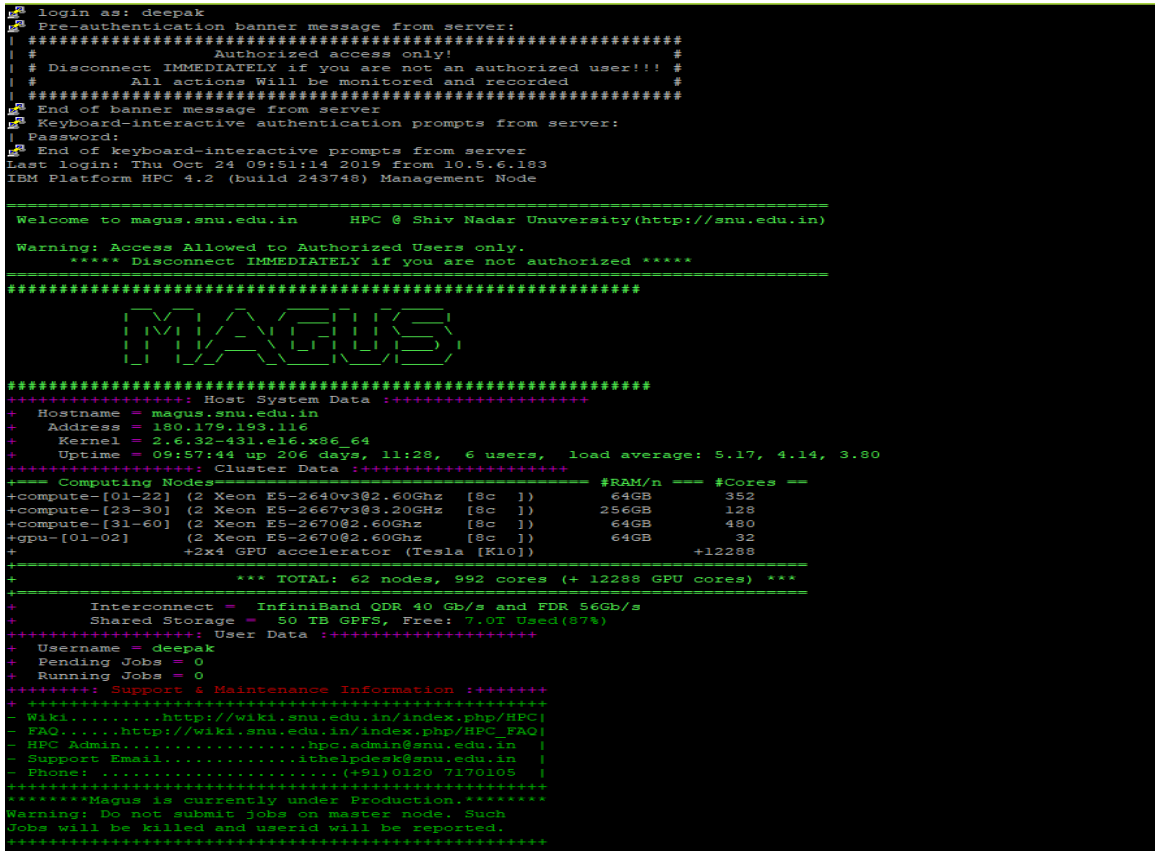



# Magus Login Screen





# Basic Linux CLI Commands



- man ( Manual )
- ls ( list directory)
- mkdir ( Make Directory )
- rmdir ( Remove Directory)
- pwd ( present working Directory)
- cp ( Copy )
- mv ( Move / Rename )
- scp ( Secure copy to other machine)
- cat ( Display contents of the file)
- tail ( Display last 10 lines)
- head ( display first 10 lines)
- chmod ( Change File permissions )
- grep ( Search within files)
- du ( Disk Usage )
- wc ( word count)

**Shortcuts**

- Ctrl+C ( kill a running command )
- Ctrl+Z ( suspend a running command)
- fg/bg ( Foreground and Background)
- Ctrl+R ( search recent commands )

- Standard Output Redirection using > and >>
- Standard Error Redirection 2 >
- Sending command to background using &
- Sending output to other command using Pipe |

# Compiling your code



- **Compiling a simple Hello World in C on linux**

- **Compiling a MPI Hello World in C on linux**

- **How to set environment variables for location of compilers and mkl libraries.**
  - *$ source /snufs/intel/parallel_studio_xe_2015/bin/psxevars.sh intel64*

# Sample Mpi Hello World

```
#include<stdio.h>
#include<mpi.h>

int main(int argc, char *argv[])
{
        int ranks, rank, n;
        int Debugmode=1;

        MPI_Init(&argc, &argv);
                char node[MPI_MAX_PROCESSOR_NAME];
                MPI_Comm_size(MPI_COMM_WORLD, &ranks);
                MPI_Comm_rank(MPI_COMM_WORLD, &rank);
                MPI_Get_processor_name(node,&n);
                printf("Greetings: %d of %d from the node %s\n", rank, ranks, node);
        MPI_Finalize();

return 0;
}
```

# Compiling and Executing



- **Setting environment variables**

  ***$ source /snufs/intel/parallel_studio_xe_2015/bin/psxevars.sh intel64***

- **Compiling**

  **$ mpiicc**

- **Executing mpi jobs**

  **$ mpirun –np <number of processors> –f <hostfile>**

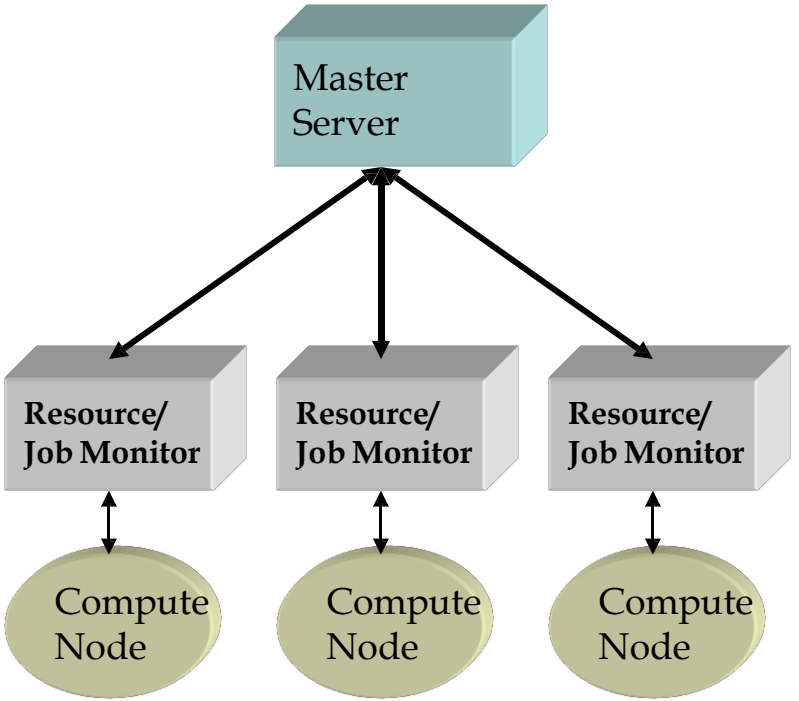
- **Mpirun options.**

  **- ppn Processor per node**

# Job Scheduler

- Allow users to share computing resources
- Utilize resources efficiently.

- IBM Platform LSF
  - distributed workload management solution for maximizing the performance of High Performance Computing (HPC) clusters.



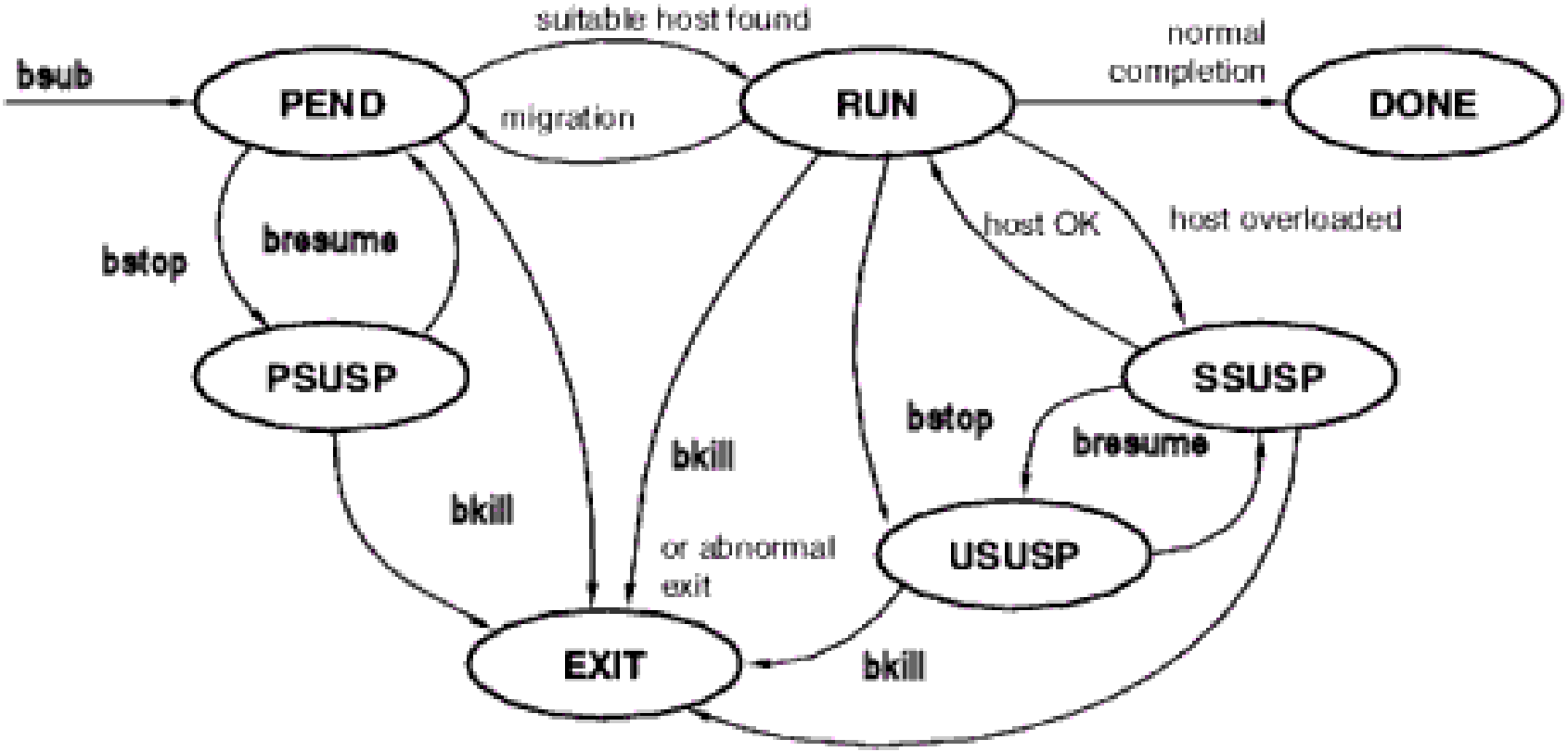
# JOB States



- LSF jobs have the following states:
  - PEND — Waiting in a queue for scheduling and dispatch
  - RUN — Dispatched to a host and running
  - DONE — Finished normally with zero exit value
  - EXIT — Finished with non-zero exit value
  - PSUSP — Suspended while pending
  - USUSP — Suspended by user
  - SSUSP — Suspended by the LSF system

# Job Transitions





# Queues



- A cluster wide container for jobs. All jobs wait in queues until they are scheduled and dispatched to hosts.
- Queues do not correspond to individual hosts; each queue can use all server hosts in the cluster, or a configured subset of the server hosts.
- When you submit a job to a queue, you do not need to specify an execution host. LSF dispatches the job to the best available execution host in the cluster to run that job.
- Queues implement different job scheduling and control policies.

# Queues on Magus



| S.No. | Processor Architecture | Priority | Queue Name | Min no of cores required to submit job | Max no of cores allowed per job | No of nodes in the queue | Wall Time | Max job per users in the Queue based on max cores |
|---|---|---|---|---|---|---|---|---|
| 1 | SandyBridge | 50 | serial_short | 1 | 1 | 1 | 1 hour | unlimited |
| 2 | SandyBridge | 50 | serial_long | 1 | 1 | 1 | 1 month | 4 |
| 3 | SandyBridge | 50 | short_sdb | 4 | 8 | 12 | 3 days | 32 |
| 4 | SandyBridge | 50 | med_sdb | 8 | 32 | 12 | 2 weeks | 32 |
| 5 | Hashwell | 50 | high_mem | 16 | 32 | 8 | 1 month | 32 |
| 6 | SandyBridge | 50 | large_sdb | 16 | 16 | 4 | 1 month | 16 |
| 7 | Hashwell | 50 | large_hsw | 16 | 16 | 4 | 1 month | 16 |
| 8 | Hashwell | 50 | med_hsw | 8 | 32 | 18 | 2 weeks | 32 |
| 9 | SandyBridge | 50 | GPU | 16 | 16 | 2 | 1 week | 16 |
| 10 | SandyBridge | 20 | long_gpu | 16 | 32 | 2 | 1 week | 16 |
| 11 | SandyBridge | 20 | long_sdb | 8 | 32 | 20 | 1 month | unlimited |
| 12 | Hashwell | 20 | long_high_mem | 8 | 32 | 8 | 1 month | unlimited |
| 13 | Hashwell | 20 | long_hsw | 8 | 32 | 20 | 1 month | unlimited |

# Working with LSF



## Job Submission

$ bsub < jobfile

```
#!/bin/bash
#BSUB -J Helloworld
#BSUB -n 16
#BSUB -q high_mem
#BSUB -e error.%J
#BSUB -o out.%J
#BSUB -u "deepak.agrawal@snu.edu.in"
#BSUB –N
#BSUB –W 10:00
#BSUB -m compute26
```

### bsub options

| | |
|---|---|
| **-B** | **Sends email when the job is dispatched** |
| **-H** | Holds the job in the PSUSP state at submission |
| **-I** | Submits a batch interactive job. |
| **-K** | Submits a job and waits for the job to finish |
| **-N** | Emails the job report when the job finishes |
| **-x** | Exclusive execution on host |
| **-b** | begin_time Dispatches the job on or after the specified date and time in the form [[month: ]day:]:minute |
| **-e** | error_file Appends the standard error output to a file |
| **-E** | "pre_exec_command[arguments...]" Runs the specified pre-exec command on the execution host before running the job |
| **-J** | "job_name" Assigns the specified name to the job. |
| **-q** | queuename Submits jobs to the specified queue |
| **-o** | output_file Appends toe standard output to a file |
| **-u** | email address < sets the email where the email has to be sent > |

# LSF Sample Submission script



```
#!/bin/bash
#BSUB -J Hello world
#BSUB -n 16
#BSUB -q high_mem
#BSUB -e error.%J
#BSUB -o out.%J
#BSUB -R "span[ptile=16]"

MPI=16  PPN=16
MYDIR=$(pwd)

EXE=/snufs/home/deepak/helloworld/a.out
OUT_FILE=$MYDIR/out.log

# Do not change anything below this
export I_MPI_DAPL_PROVIDER=ofa-v2-mlx4_0-1
export I_MPI_FABRICS=shm:dapl
export I_MPI_FALLBACK=0
export OMP_NUM_THREADS=1
export FORT_BUFFERED=yes
export I_MPI_PIN_PROCESSOR_LIST=0-15
rm -f host.list
cat $LSB_DJOB_HOSTFILE > ./host.list  env > log.env
mpiexec.hydra -np $MPI -f ./host.list -genvall -ppn $PPN $EXE 2>&1 | tee -a $OUT_FILE
```

# Working with LSF – JOB Administration



- **Kill a running Job**
  - **$ bkill < jobid>**
- **See the status of your jobs in the Queue**
  - **$ bjobs**
- **See status of all user jobs**
  - **$ bjobs –u all**
- **See Information about Queues**
  - **$ bqueues**
- **See Resources on hosts**
  - **$ bhosts**

# Software's and Libraries - VASP



- **Environment Variables in .bashrc**
  - source /snufs/intel/composer_xe_2015.2.164/bin/compilervars.sh intel64
  - source /snufs/intel/impi/5.0.3.048/intel64/bin/mpivars.sh
- **Location of VASP-5.3.5 Binaries.**
  - **Vasp for Hashwell**
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.hw
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.hw.gamma
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.hw.nc
  - **Vasp binaries for Sandybridge**
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.sdb
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.sdb.gamma
    - /snufs/apps/vasp/5.3.5/intelmpi/vasp.sdb.nc
- **Sample Job Script :** */snufs/apps/vasp/5.3.5/intelmpi/submit.lsf.example*

# Applications



- Gromacs
- Vasp
- Quantum espresso
- BigDFT
- Lammps
- Gaussian
- USPEX
- Matlab ( single node)

# Thank You



Deepak Agrawal

deepak.agrawal@snu.edu.in

Extn : 120

9560084920