High Performance Computing (HPC) - Tutorial Session

Bocconi Students for Machine Learning

Giacomo Ciro’
Vice-president & Co-founder

High Performance Computing (HPC) Cluster

1

What Is It

Layout & Specs

2

How to Access

Asking Permission

Using the VPN

3

How to Use

Submit, Monitor & Cancel Jobs

Basic Linux

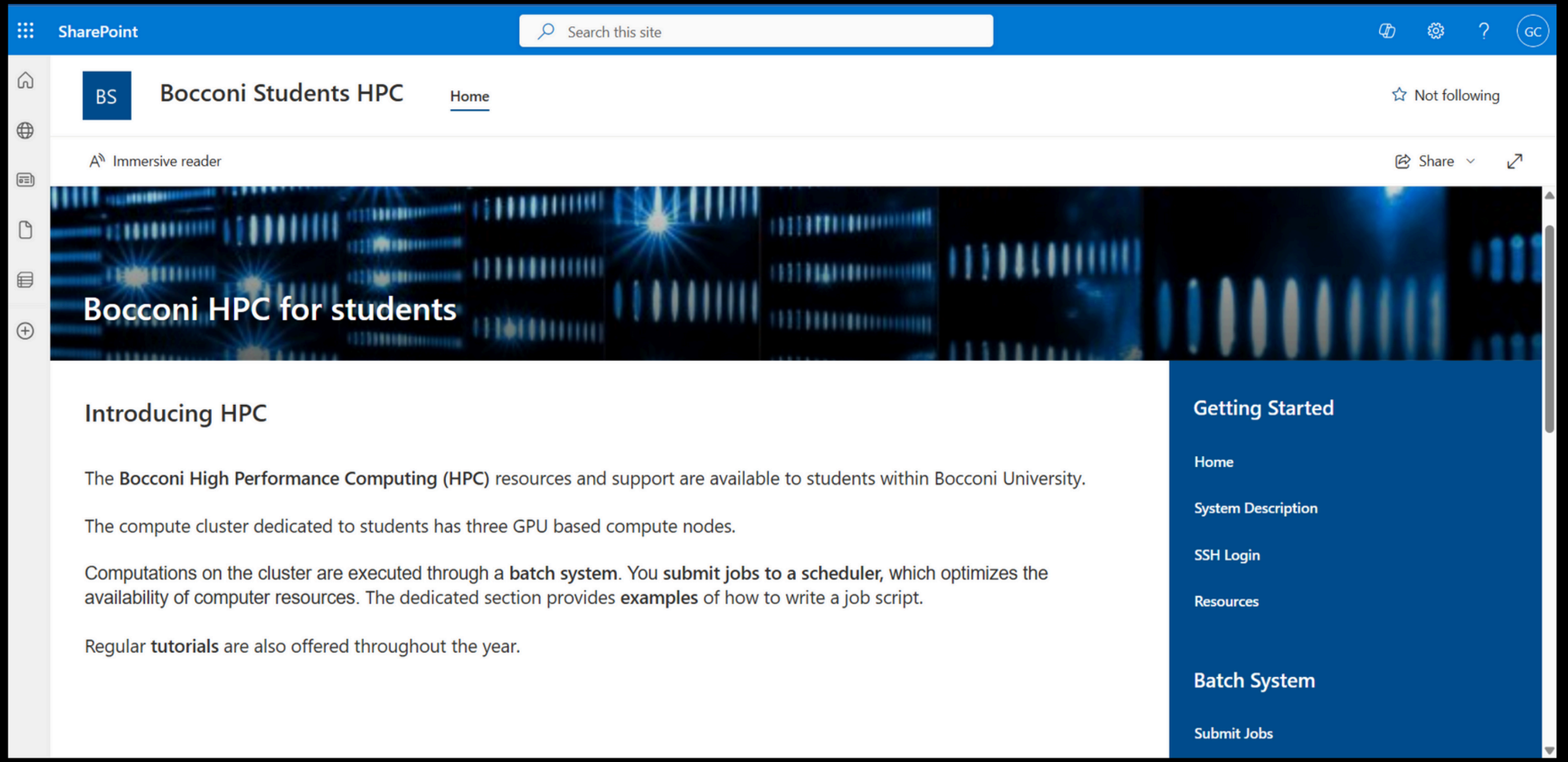
4

Tutorial

Hands-on practice

Online Resources

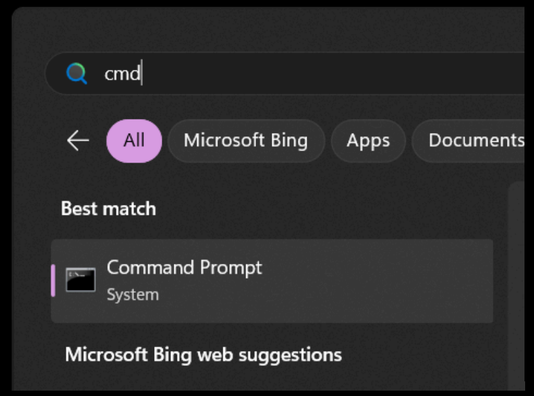
https://bocconi.sharepoint.com/sites/BocconiStudentsHPC

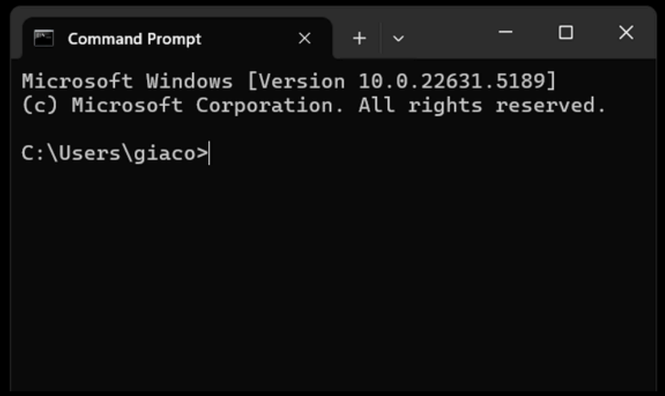


Background Knowledge

Terminal

- a command-line interface (CLI) to interact with a computer in plain text (text input / text output)
- the actual program interpreting the commands is called *shell*
- in Unix-like systems (e.g. Linux, MacOS), this is *bash* (stands for Bourne Again Shell, because it's an improvement of the previous shell by Stephen Bourne)

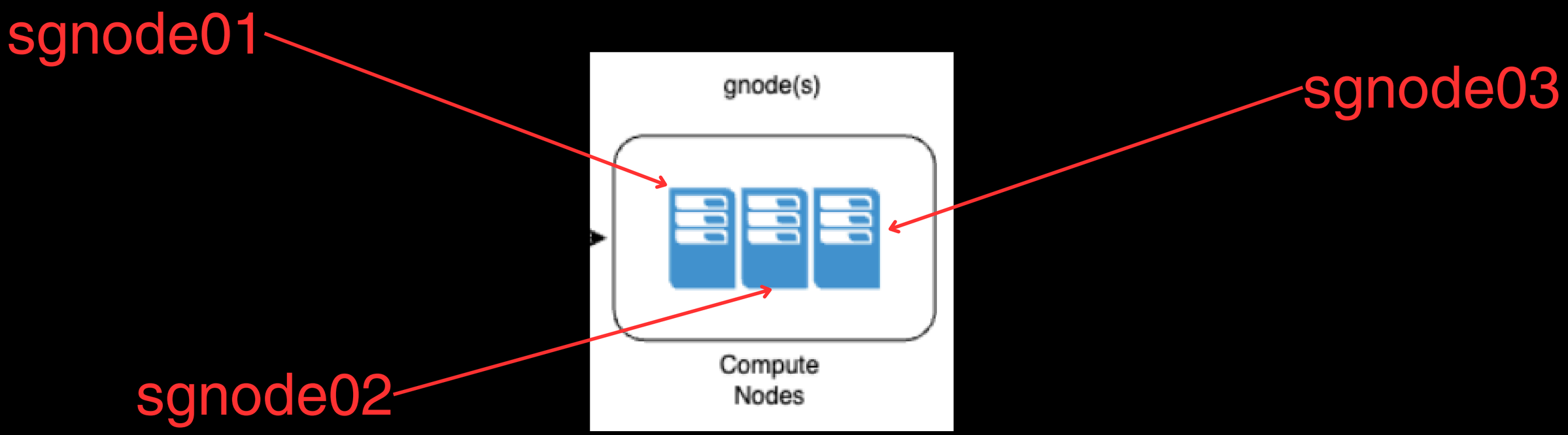




Anaconda Distribution

- open-source Python distribution of specifically designed for data science, with
- nice package manager (*conda*)
- it's a huge snake (use *miniconda*)

1 What Is It



sgnode = ScaleGrid Node (grid computing, distributed computing)
slnode = ScaleLogin Node (loging and submit jobs to the grid)

1 Specs

sgnode	partition	CPU	RAM	GPU
01	stud / ai	Intel i7-7820X (16 cores)	23 GB	2 x nVidia 1080 Ti (2 x 11GB GDDR5)
02	??	??	102 GB	4 x nVidia 2080 Ti (4 x 11GB GDDR6)
03	dsba	2 x Intel Xeon Platinum 8160 (96 cores)	495 GB	3 x nVidia 2080 Ti (3 x 11GB GDDR6)

1 Specs

```
NodeName=sgnode01 Arch=x86_64
   CPUAlloc=0 CPUEfctv=16 CPUT
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:nv-1080:2
   NodeAddr=sgnode01 NodeHostN
   OS=Linux 4.18.0-513.18.1.el
   RealMemory=23000 AllocMem=0
   State=IDLE ThreadsPerCore=2
   Partitions=stud,ai
```

```
NodeName=sgnode02 CoresPerSo
   CPUAlloc=0 CPUEfctv=16 CP
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:nv-2080:4
   NodeAddr=sgnode02 NodeHos
   RealMemory=102400 AllocMe
   State=UNKNOWN+NOT_RESPOND
   BootTime=None SlurmdStart
   LastBusyTime=2025-03-24T1
```

```
NodeName=sgnode03 Arch=x86_64
   CPUAlloc=18 CPUEfctv=96 CPU
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:nv-2080:3
   NodeAddr=sgnode03 NodeHostN
   OS=Linux 4.18.0-513.18.1.el
   RealMemory=495000 AllocMem=
   State=MIXED ThreadsPerCore=
   Partitions=dsba
```

GNode01 Architecture: Pascal

- Motherboard Asus WS-X299-SAGE
- cpu Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz
- 32 GB RAM
- 3 x NVIDIA GeForce RTX 2080 Ti
- 1 x NVME 1TB

GNode02 Architecture: Pascal

- Motherboard Asus WS-X299-SAGE
- 64 GB RAM
- 4 x NVIDIA GeForce RTX 1080 Ti
- 1 x NVME 1TB

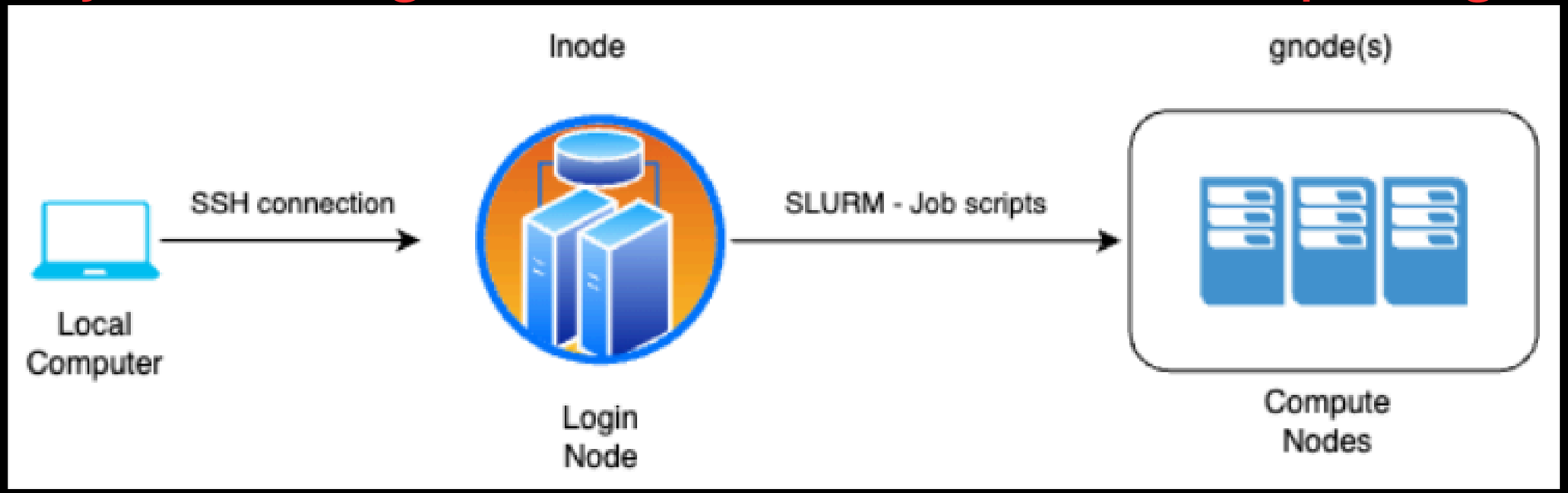
GNode03 Architecture: Pascal

- Motherboard Intel S2600STB
- 2 cpu Intel Xeon Platinum 8160 CPU 2.10GHz
- 512 GB RAM
- GPU 3 x NVIDIA TITAN Xp
- 1 x NVME 500 GB

2 How to Access

login, file storage & job management

actual computing



2 How to Access

Secure Shell (SSH)

- cryptographic network protocol enabling secure remote login and command execution between computers

Simple Linux Utility for Resource Management (SLURM)

- open-source workload manager system for Linux environments
- used to allocate resources and manage job executions

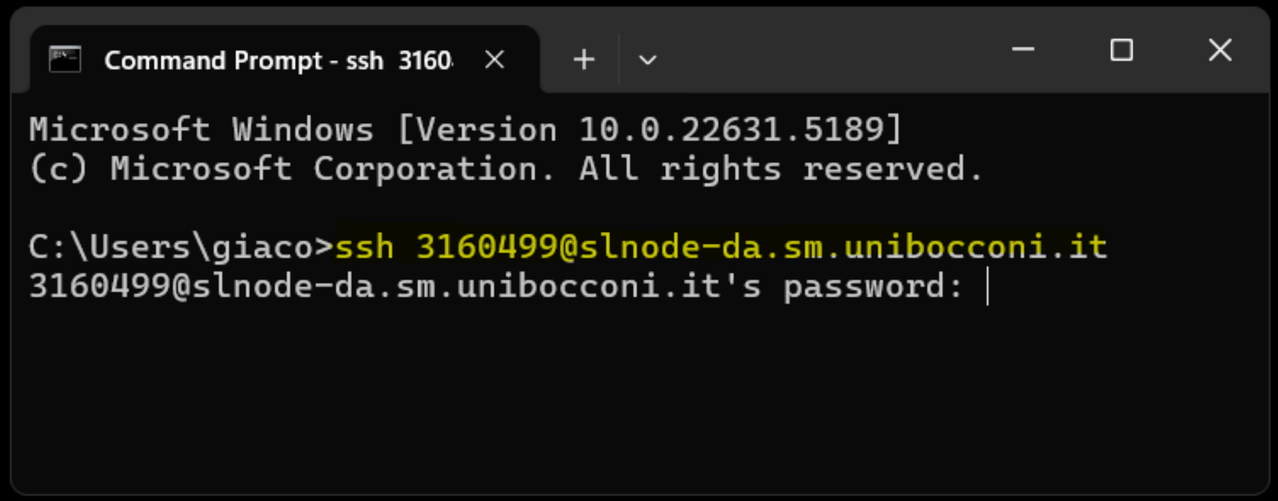
2 ssh to the cluster

Interactions with the HPC are made thorugh the terminal.

You use ssh protocol to access the HPC remotely and do stuff:

`ssh` *`bocconi_id`*`@slnode-da.sm.unibocconi.it`

And insert your password (the one for You@B).

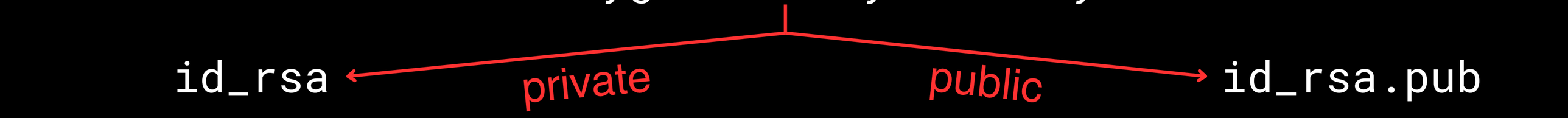


2 ssh keys

To avoid typing the password every time, you can setup SSH keys.

On the local machine, you create a pair of private and public keys using:

```
ssh-keygen -C “my_new_key”
```




```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAA
Ebm9uZQAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACCDpG/yUk+mv/82pDOgjE9VC
m1ZOxcF1ru+acx/2Hz/QQAAAJA6owi3OqMI
twAAAAtzc2gtZWQyNTUxOQAAACCDpG/yUk+
mv/82pDOgjE9VCm1ZOxcF1ru+acx/2Hz/QQ
AAAEDpDpt3s2dPJDxGrNaNvXJd8CXJ+B1D/
tq+N69r9iGjy4Okb/JST6a//zakM6CMT1UK
bVk7FwXWu75pzH/YfP9BAAAACm15X25ld19
rZXkBAgM=
-----END OPENSSH PRIVATE KEY-----
```

keep on **local**/path/to/.ssh

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIIO
kb/JST6a//zakM6CMT1UKbVk7Fw
XWu75pzH/YfP9B my_new_key
```

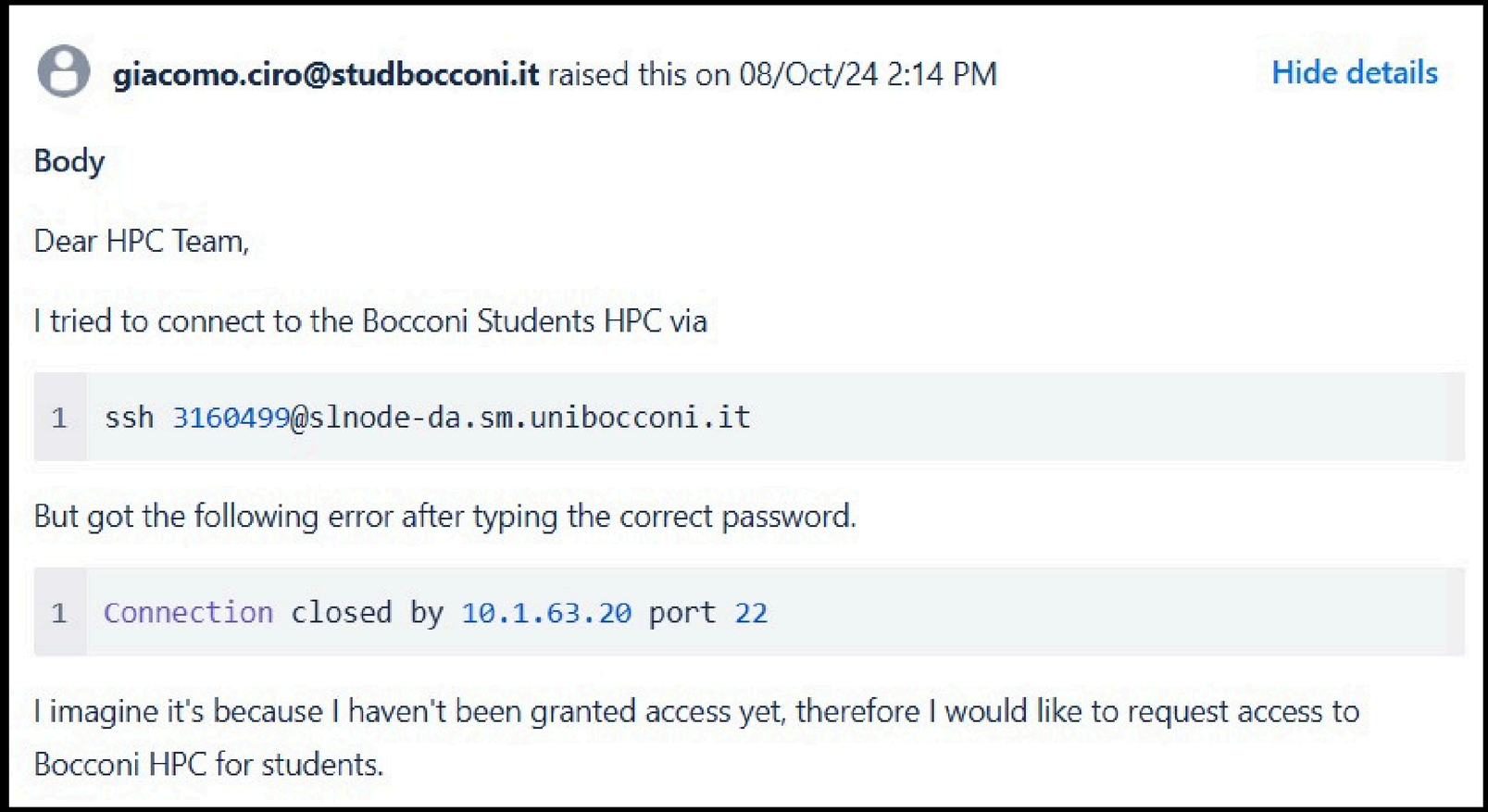
copy-paste into
hpc/path/to/.ssh/authorized_keys

2 Request permission

If it’s the first time, you need to ask to activate your account.

Send an email to:

hpc@unibocconi.it



giacomo.ciro@studbocconi.it raised this on 08/Oct/24 2:14 PM Hide details

Body

Dear HPC Team,

I tried to connect to the Bocconi Students HPC via

```
1  ssh 3160499@slnode-da.sm.unibocconi.it
```

But got the following error after typing the correct password.

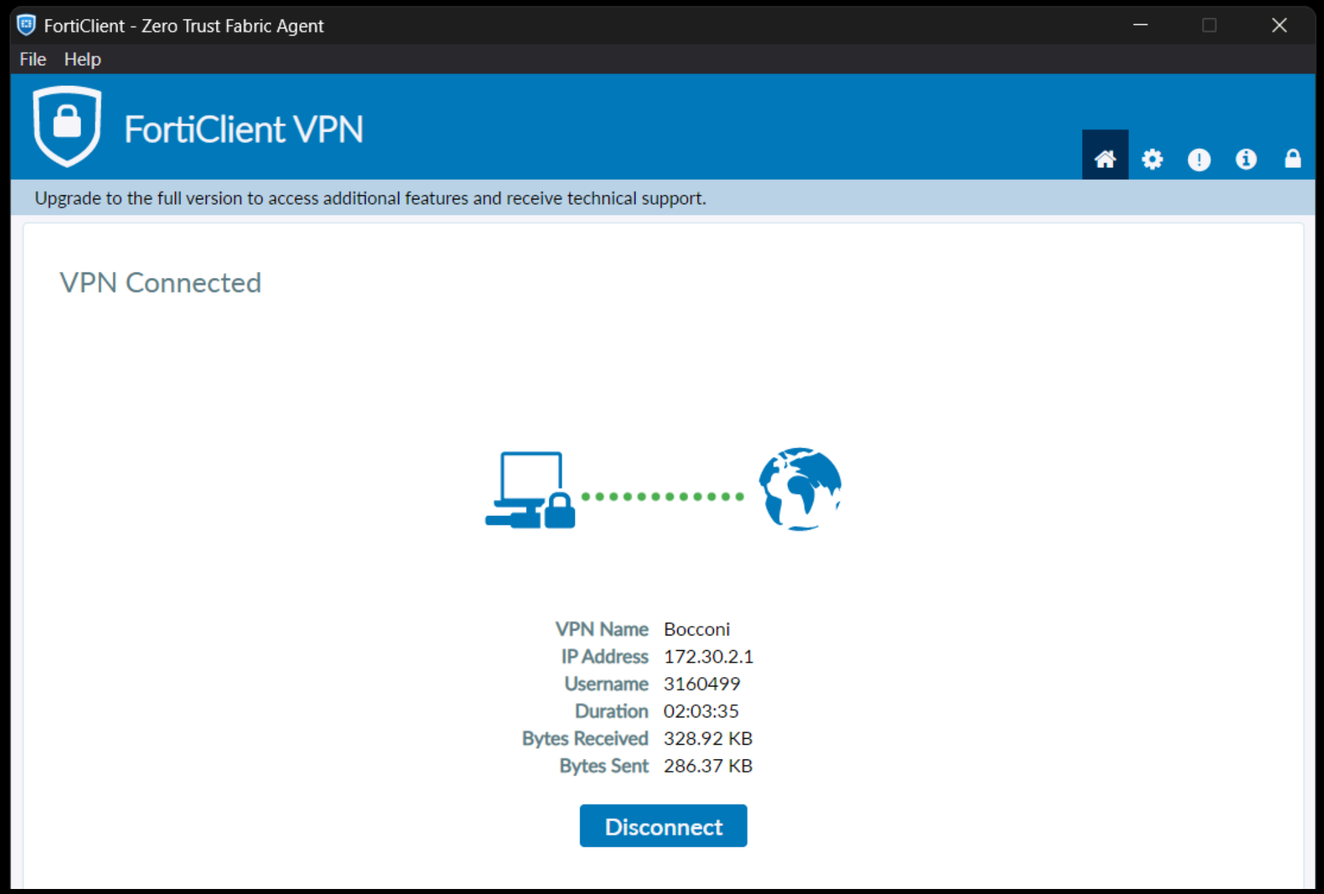
```
1  Connection closed by 10.1.63.20 port 22
```

I imagine it's because I haven't been granted access yet, therefore I would like to request access to Bocconi HPC for students.

2 Using the VPN

You must be connected to the bocconi network:

- physically on campus
- via VPN



https://www.unibocconi.it/en/current-students/technology-and-information-services/access-bocconi-network-through-vpn

2 Using the VPN

1) Download the Fortinet SSL VPN client:

Windows https://download.unibocconi.it/ForticlientSSLVPN/FortiClientVPNOnlineInstaller.exe
MacOS https://download.unibocconi.it/ForticlientSSLVPN/FortiClientVPNOnlineInstaller_7.0.dmg
Linux (.deb) https://download.unibocconi.it/ForticlientSSLVPN/forticlient_vpn_7.0.2.0063_amd64.deb
Linux (.rpm) https://download.unibocconi.it/ForticlientSSLVPN/forticlient_vpn_7.0.2.0063_x86_64.rpm

2) Log-in using You@B credentials (student id and password)

3) Specify the VPN settings:

Connection Name: Bocconi
Remote Gateway: vpn.unibocconi.it
Customize port: 443

2 MacOS VPN

Additonal trick required to setup VPN on MacOS:

Settings
> Login Items & Extensions
> Network Extension
> Allow VPN Provider *fortitray.app*

Credits: Davide Beltrame

3 How to Use

Computations on the cluster are performed by submitting a SLURM job in the form of a bash script (a file named jobname.sh):

```
sbatch job.sh
```

Each SLURM job specifies the actions the computer should perform (usually, it is just “run this python script”).

After the job is submitted, SLURM takes care of executing it as soon as the required resources are available.

GNU nano 2.9.8 job.sh Modified

job.sh

```
#!/bin/bash
#SBATCH --job-name="test"
#SBATCH --account=3160499
#SBATCH --partition=ai

#SBATCH --cpus-per-task=1
#SBATCH --gpus=1
#SBATCH --mem=10GB

#SBATCH --chdir=.
#SBATCH --output=/home/3160499/out/%x_%j.out
#SBATCH --error=/home/3160499/err/%x_%j.err

#SBATCH --mail-type=ALL
#SBATCH --mail-user=giacomo.ciro@unibocconi.it

module load modules/miniconda3
eval "$(conda shell.bash hook)"

conda activate bsml

python script.py

module unload modules/miniconda3

echo "The end"
```

Header

- job metadata
- resources required
- logging info

Body

- the actual actions to perform

3 Header

```
#!/bin/bash
#SBATCH --job-name="test"
#SBATCH --account=3160499
#SBATCH --partition=ai

#SBATCH --cpus-per-task=1
#SBATCH --gpus=1
#SBATCH --mem=10GB

#SBATCH --chdir=.
#SBATCH --output=/home/3160499/out/%x_%j.out
#SBATCH --error=/home/3160499/err/%x_%j.err

#SBATCH --mail-type=ALL
#SBATCH --mail-user=giacomo.ciro@unibocconi.it
```

stud / ai / dsba

the working directory

%x = job name
%j = job id

NONE, BEGIN, END, FAIL, REQUEUE, ALL, TIME_LIMIT, ARRAY_TASKS

3 Body

```
module load modules/miniconda3
eval "$(conda shell.bash hook)"

conda activate bsml

python script.py

module unload modules/miniconda3

echo "The end"
```

required to use miniconda

activate the environment

run the script

final housekeeping

3 SLURM cmds

`sbatch job.sh` (submit a job)

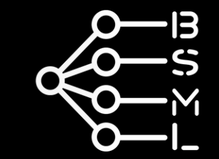
```
(base) [3160499@slnode ~]$ sbatch job.sh
Submitted batch job 4797
(base) [3160499@slnode ~]$
```

`squeue` (show priority queue, now useless)

```
(base) [3160499@slnode ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              4798        ai     test  3160499  R       0:03      1 sgnode01
(base) [3160499@slnode ~]$
```

`scancel` (cancel a job)

```
(base) [3160499@slnode ~]$ scancel 4799
(base) [3160499@slnode ~]$
```



3 SLURM cmds

`sinfo` (show nodes info)

```
(base) [3160499@slnode ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
stud         up    4:10:00      1   idle sgnode01
dsba         up   20:10:00      1    mix sgnode03
ai           up   20:10:00      1   idle sgnode01
(base) [3160499@slnode ~]$
```

`sacct` = slurm accounting (show past jobs)

```
(base) [3160499@slnode ~]$ sacct
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
4797             test         ai    3160499          1  COMPLETED      0:0
4797.batch      batch              3160499          1  COMPLETED      0:0
4798             test         ai    3160499          1  COMPLETED      0:0
4798.batch      batch              3160499          1  COMPLETED      0:0
4799             test         ai    3160499          1 CANCELLED+      0:0
4799.batch      batch              3160499          1  CANCELLED     0:15
(base) [3160499@slnode ~]$
```

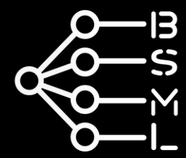
3 SLURM cmds

`scontrol show node <node_name>` (node specific info)

```
(base) [3160499@slnode ~]$ scontrol show node sgnode01
NodeName=sgnode01 Arch=x86_64 CoresPerSocket=8
   CPUAlloc=0 CPUEfctv=16 CPUTot=16 CPULoad=0.00
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:nv-1080:2
   NodeAddr=sgnode01 NodeHostName=sgnode01 Version=23.11.4
   OS=Linux 4.18.0-513.18.1.el8_9.x86_64 #1 SMP Thu Feb 1 03:51:05 EST 2024
   RealMemory=23000 AllocMem=0 FreeMem=4821 Sockets=1 Boards=1
   State=IDLE ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
```

`scontrol show partition <partition_name>` (partition specific info)

```
(base) [3160499@slnode ~]$ scontrol show partition stud
PartitionName=stud
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=04:10:00 MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED MaxCPUsPerSocket=UNLIMITED
   Nodes=sgnode01
   PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=16 TotalNodes=1 SelectTypeParameters=NONE
   JobDefaults=(null)
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
   TRES=cpu=16,mem=23000M,node=1,billing=16,gres/gpu=2
```



3 LINUX cmds

`ls path/to/dir` (list directory content)

```
(base) [3160499@slnode ~]$ ls
archive          f1-speed-estimation-bench  speed.sh
diffusion-llms   job.sh                     test.py
diffusion.sh     out.out
err.err          OUTpy
(base) [3160499@slnode ~]$
```

`cat path/to/file` (print file content)

```
(base) [3160499@slnode ~]$ cat job.sh
#!/bin/bash
#SBATCH --job-name="test"
#SBATCH --account=3160499
#SBATCH --partition=ai

#SBATCH --cpus-per-task=1
```

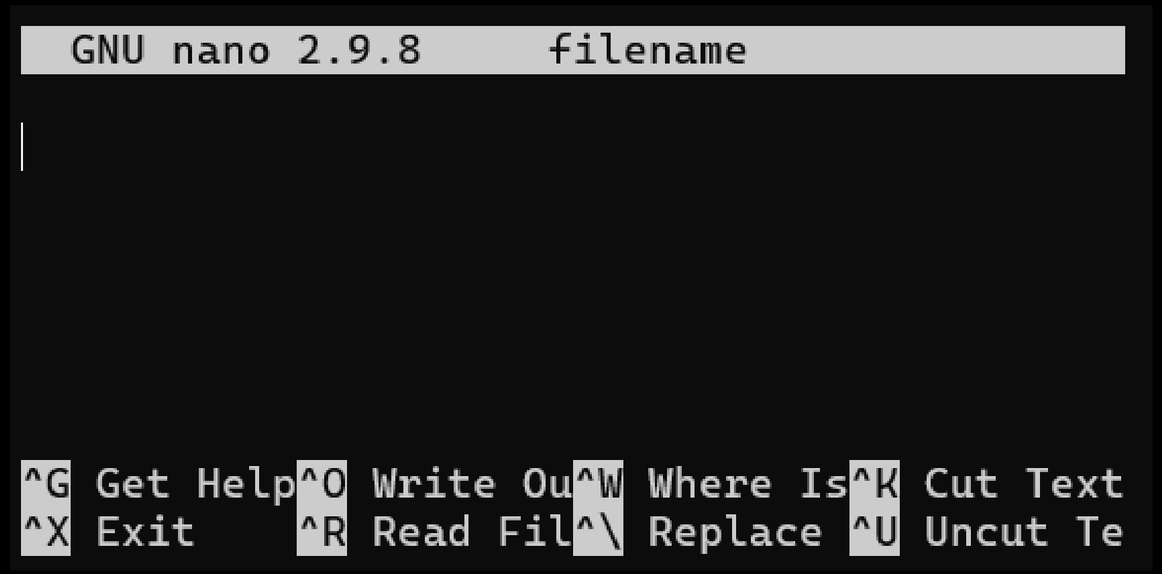
3 LINUX cmds

`rm path/to/file` (remove file, -r to remove directory)

```
(base) [3160499@slnode ~]$ rm -r /mnt/c
```

`nano path/to/file` (minimalist text editor)

```
(base) [3160499@slnode ~]$ nano filename
```



3 LINUX cmds

`tail -f path/to/file` (show end of file, refreshes when changes occur)

```
(base) [3160499@slnode ~]$ tail -f err.err
slurmstepd: error: *** JOB 4799 ON sgnode01 CANCELLED AT 2025-04-13T16:27:23 ***
```

ctrl + c to exit

`free -h` (memory info)

```
(base) [3160499@slnode ~]$ free -h
               total        used        free      shared  buff/cache   available
Mem:           7.8Gi       564Mi       1.9Gi       0.0Ki       5.3Gi       6.9Gi
Swap:          2.0Gi       500Mi       1.5Gi
(base) [3160499@slnode ~]$
```

`watch -n 1 <cmd>` (run command every 1 second)

```
(base) [3160499@slnode ~]$ watch -n 1 date
```

```
Every 1.0s: date

Sun Apr 13 17:26:34 CEST 2025
```

LINUX cmds

`nvidia-smi` (GPU info)

```
Sun Apr 13 16:27:19 2025
+-----------------------------------------
| NVIDIA-SMI 550.54.14              Driver Version
|-----------------------------------------+-------
| GPU  Name                 Persistence-M | Bus-Id
| Fan  Temp   Perf          Pwr:Usage/Cap |
|                                         |
|=========================================+=======
|   0  NVIDIA GeForce GTX 1080 Ti     On  |   0000
| 30%   50C    P8              13W /  280W |
|                                         |
+-----------------------------------------+-------

+-----------------------------------------
| Processes:
|  GPU   GI   CI        PID   Type   Process name
|        ID   ID
|=========================================
|  No running processes found
+-----------------------------------------
```

`lscpu` (CPU info)

```
(base) [3160499@slnode ~]$ lscpu
Architecture:            x86_64
CPU op-mode(s):          32-bit, 64-bit
Byte Order:              Little Endian
CPU(s):                  4
On-line CPU(s) list:     0-3
Thread(s) per core:      1
Core(s) per socket:      1
```

3 Misc

- Virtual environments are managed by *Conda* and are created directly on the login node with the usual commands:

```
conda create -n myenv
conda activate myenv
conda install <pkg>
python -m pip install <pkg>
```

- Use GitHub to transfer code to the HPC

- Use VSCode to connect to the HPC using the nice GUI

4 Tutorial

Let’s try to run our first python script on the HPC cluster!

Thank You!

Bocconi Students for Machine Learning

Giacomo Ciro’
Vice-president & Co-founder