جامعـــة خليفـــة
Khalifa University

Almesbar Introductory Session

Dr. Sergio Martinez

What is the HPC cluster? – Almesbar specifications

- **Number of nodes**
- Cores and memory
- Network and storage
- Performance

Number of nodes	
88 CPU nodes	2x Intel Xeon Gold 6230R 26-Core 2.1GHz (Cascade lake) (Q1‘20)
3 GPU nodes	2x Intel Xeon Gold 6230R 26-Core 2.1GHz(Cascade lake) (Q1’20)
	4x NVidia V100

What is the HPC cluster? – Almesbar specifications

- Number of nodes
- **Cores and memory**
- Network and storage
- Performance

Cores per node	
52	
Memory per node	
CPU nodes	384 GB
GPU nodes	640 GB + 128 GB (GPU)

Total CPU cores
4,732
Total GPU cores
61,440
Total memory
35.25 TB

What is the HPC cluster? – Almesbar specifications

- Number of nodes
- Cores and memory
- **Network and storage**
- Performance

Network interconnection	
Infiniband EDR (100Gb/s) and Infiniband HDR (200Gb/s)	
Storage	
300 TB home	NFSv4 (10Gb/s)
947 TB scratch	Parallel file system (Lustre) (200Gb/s)

What is the HPC cluster? – Almesbar specifications

- Number of nodes
- Cores and memory
- Network and storage
- **Performance**

Theoretical peak of performance (CPU)
307.5 TFOPS
Achieved peak of performance (CPU)
205.0 TFOPS

Almesbar - Accounts, resource allocation and file sharing

- **User name and access**
- HPC Projects
- Jobs
- Queues
- Storage
- License applications

You will be using your KU Employee ID as your user name for accessing the HPC cluster.

You will be using SSH keys to identify yourself and access the HPC cluster.

SSH keys allow secure, password-less access to remote systems. They are very easy to set up and provide a greater level of security than using a password, while being convenient too.

We will discuss how to create and use your keys in the coming sections.

Almesbar - Accounts, resource allocation and file sharing

- User name and access
- **HPC Projects**
- Jobs
- Queues
- Storage
- License applications

Resource allocation and file sharing on Almesbar HPC cluster is based on HPC projects.

Every single user account will be associated with at least one HPC project.

An HPC project will also be a shared space where project members can share files and submit their jobs.

Ultimately, every job has to be accounted against an HPC project.

Fairshare is applied to the HPC projects at first level.

Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- **Jobs**
- Queues
- Storage
- License applications

Jobs number:

- Limit of 1000 jobs per user on **prod** and **gpu** queues
- Limit of 1 job per user on **devel** queue

Jobs size:

- Wall time limitations on each queue.
- There is no limit on the number of cores used by a job.

Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- **Queues**
- Storage
- License applications

Access is restricted for train queue.

Research computing department uses **train** queue for compiling, installing, maintaining applications as well as for troubleshooting with the HPC users.

Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- Queues
- **Storage**
- License applications

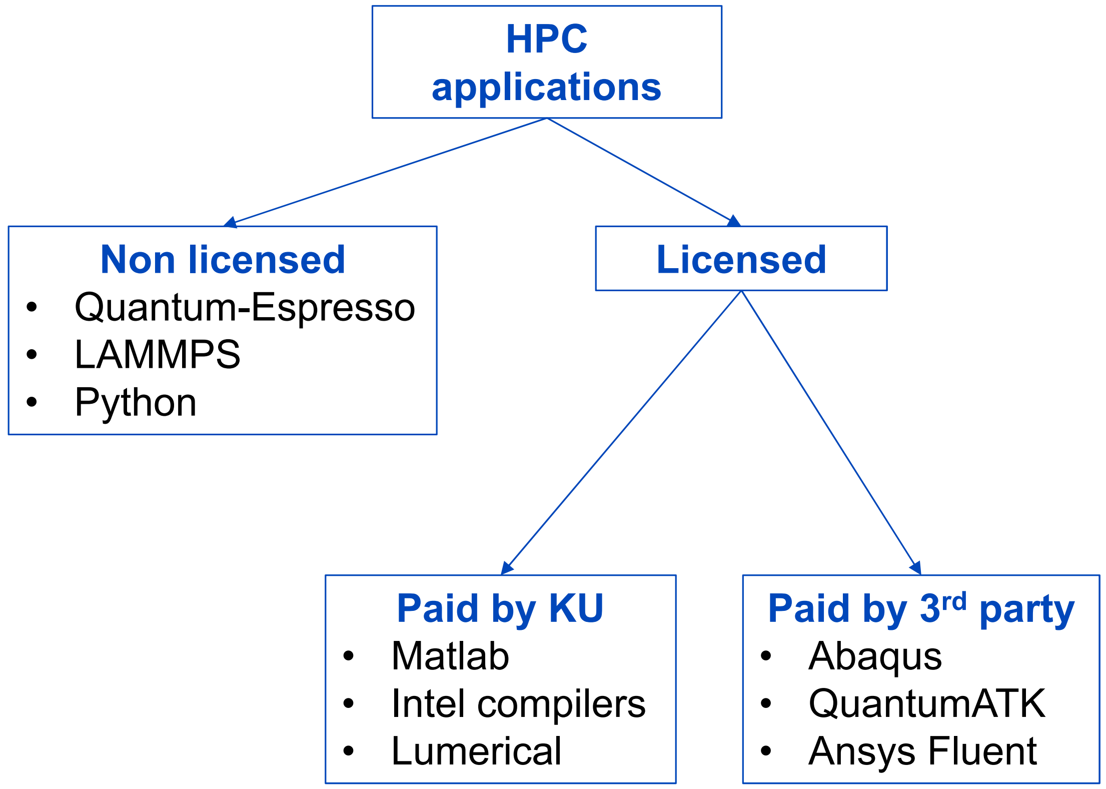
There will be quotas applied to the different storage places.

By default:

- Home directory (`/home/kunet.ae/username`): 50 GB
- Project directory (`/l/proj/projectname`): 2 TB

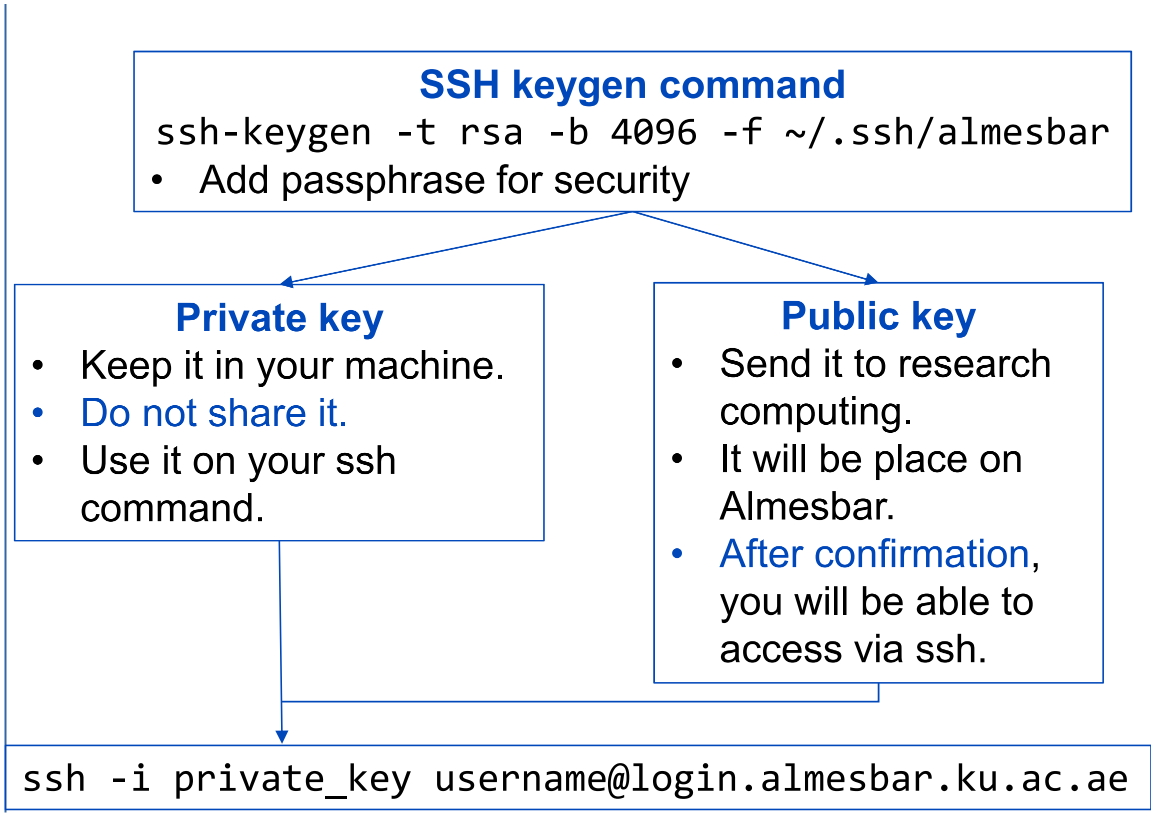
Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- Queues
- Storage
- **License applications**



Almesbar configuration

- **Access configuration**
- Queues (partitions)
- Modular environment
- Scheduler syntax



SSH keygen command

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/almesbar
```

- Add passphrase for security

Private key

- Keep it in your machine.
- Do not share it.
- Use it on your ssh command.

Public key

- Send it to research computing.
- It will be place on Almesbar.
- After confirmation, you will be able to access via ssh.

```
ssh -i private_key username@login.almesbar.ku.ac.ae
```

Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

Production	
Name	prod
Purpose	Production jobs using CPU cores.
Number of nodes	80
Limits	Default wall time: 2h Maximum wall time: 48h Max number of jobs per user: 1000
Access	Open

Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

Development	
Name	devel
Purpose	For interactive and testing purposes, i.e, compiling, configuring and testing applications. Running GUI applications.
Number of nodes	4
Limits	Default wall time: 1h Maximum wall time: 8h Max number of jobs per user: 1
Access	Open

Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

GPU	
Name	gpu
Purpose	Production jobs using GPU cores.
Number of nodes	3
Number of GPUs per node	4
Limits	Default wall time: 2h Maximum wall time: 48h Max number of jobs per user: 1000
Access	Open

Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

Training	
Name	train
Purpose	Used by Research Computing department for compiling, installing, maintaining applications as well as for troubleshooting with the HPC users.
Number of nodes	4
Limits	No
Access	Restricted

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

On a complex computer system, on which it is necessary to make available a wide choice of software packages in multiple versions, it can be quite difficult to set up the user environment so as to always find the required executables and libraries.

This is particularly true where different implementations or versions use the same names for files.

Environment modules provide a way to selectively activate and deactivate modifications to the user environment which allow particular packages and versions to be found.

Our modular environment is hierarchical and it has three levels:

Core, Compiler and MPI

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

Useful commands:

Command	Example	Description
module avail	module avail	Lists available modules in the system. (D) means default. (L) means it is already loaded. Press Q to exit the list at any moment.
module list	module list	List your currently loaded modules.
module load module_name	module load matlab/R2021a	Loads the module for Matlab version R2019b
module show module_name	module show matlab/R2021a	Displays information about one or more module files.
module spider string	module spider wrf	List all module that contain the "string“ with details on how to load them.
module swap old_module new_module	module swap python/2.7.3 python/3.6.0	Switches similar/conflicting modules

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

Core level:

- Modules that are not dependent on Compiler or MPI implementation.
- Most of the installed applications that don't require to be compiled seats in this level. Also the compilers can be found in this level.
- This is the only level that is visible for `module avail` command after logging in the system.
- Only if user loads a module that is a compiler, for example `gcc/9.3` , the next level in the hierarchy (Compiler) will be available and became visible with module avail.

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

Core level:

```
[username@login-3 ~]$ module avail
-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019           cuda/11.3               miniconda/3
   abaqus/2021    (D)    gcc/9.3                 molpro/2021.1.0
   ansys/2021R1          lumerical/2021R1        starccm/16.02.009
   ansysem/2021R1        matlab/R2020b
   comsol/56             matlab/R2021a    (D)
```

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

Compiler level:

- This level becomes available only if the user loads a compiler module, like `gcc/9.3` from the Core level.
- Then, all the applications compiled with that compiler (`gcc/9.3`) that don't require MPI to be compiled, will be listed here. This include MPI implementations like `openmpi` or `mvapich` for example.
- Only if user loads a module that is an MPI implementation, for example `openmpi/4.0` , the next level in the hierarchy (MPI) will be available and became visible with `module avail` command.

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

Compiler level:

```
[username@login-3 ~]$ module load gcc/9.3
[username@login-3 ~]$ module avail

-------------------- /apps/ku/modulefiles/Compiler/gcc/9.3 --------------------
   hdf5/1.10.6        libpng/1.2.12     ucx/1.8.0
   jasper/1.701.0     openblas/0.3.7    wannier90/3.1.0
   libfabric/1.10.1   openmpi/4.0       zlib/1.2.11

-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019             cuda/11.3                 miniconda/3
   abaqus/2021    (D)      gcc/9.3            (L)    molpro/2021.1.0
   ansys/2021R1            lumerical/2021R1          starccm/16.02.009
   ansysem/2021R1          matlab/R2020b
   comsol/56               matlab/R2021a      (D)
```

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

MPI level:

- This level becomes available only if the user loads an MPI implementation module, like `openmpi/4.0` from the Compiler level.
- Then, all the applications and libraries compiled with that compiler/MPI pair (`gcc/9.3` and `openmpi/4.0` will be listed here.

Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

MPI level:

```
[username@login-3 ~]$ module load openmpi/4.0
[username@login-3 ~]$ module avail

---------------- /apps/ku/modulefiles/MPI/gcc/9.3/openmpi/4.0 ----------------
   elpa/2021.05.001        netcdf/4.7.3       vasp/6.2.0-wannier90-3.1.0
   fftw/3.3.8              phdf5/1.10.6       wps/4.0
   lammps/29Oct2020        pnetcdf/1.12.1     wrf/4.0
   netcdf-cxx/4.3.1        qe/6.7.0
   netcdf-fortran/4.5.2    scalapack/2.1.0

-------------------- /apps/ku/modulefiles/Compiler/gcc/9.3 --------------------
   hdf5/1.10.6         libpng/1.2.12     ucx/1.8.0
   jasper/1.701.0      openblas/0.3.7    wannier90/3.1.0
   libfabric/1.10.1    openmpi/4.0  (L)  zlib/1.2.11

-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019             cuda/11.3               miniconda/3
   abaqus/2021    (D)      gcc/9.3          (L)    molpro/2021.1.0
   ansys/2021R1            lumerical/2021R1        starccm/16.02.009
   ansysem/2021R1          matlab/R2020b
   comsol/56               matlab/R2021a    (D)
```

Almesbar configuration

- ➢ Access configuration
- ➢ Queues (partitions)
- ➢ Modular environment
- ➢ **Scheduler syntax**

Job submission control

Command	Example	Description
salloc	salloc -n 1	Obtain a job allocation.
sbatch	sbatch my_script.sh	Submit a batch script for later execution.
srun	srun -n 1 executable	If run from command line directly, srun will first create a resource allocation in which to run the parallel job. srun can be run from within sbatch script or salloc shell
sinfo	sinfo -p prod	Display partition information
	sinfo -N	Display nodes information
squeue	squeue -p prod	Display information about jobs
scontrol	scontrol show job 3420	Show active job details
sacct	sacct -j 3420	Show accounting information about active and completed jobs
scancel	scancel 3420	Send a signal to kill the job

Almesbar configuration

- ➢ Access configuration
- ➢ Queues (partitions)
- ➢ Modular environment
- ➢ **Scheduler syntax**

Configuring a job

Option	Example	Description
#SBATCH		Scheduler directive to be used on job configuration scripts
--nodes	--nodes=2	Will request tasks be run across 2 nodes
--ntasks	--ntasks=2	Will start 2 MPI tasks
--ntasks-per-node	--ntasks-per-node=1	Will start 1 task per requested node
--cpus-per-task	--cpus-per-task=10	Will request 10 cores per task
--account	--account=project_name	The project your core hours will be 'charged' to.
--partition	--partition=prod	Submits job the specified partition. On Almesbar HPC cluster, the default queue is “prod"
--output	--output=file_name	Writes job output to “file_name”, %j can be used to include the JOBID, e.g. result.%j.out
--error	--output=file_name	Writes job errors to “file_name”, %j can be used to include the JOBID, e.g. result.%j.err
--job-name	--job-name=job_name	Assigns the specified name “job_name” to the job
--time	--time=DD-HH:MM:SS	Requests wall time limit
--mem	--mem=512MB	Requests a specific amount of memory for your job
--x11	salloc -n 1 --x11 srun -n 1 --x11 xterm	Enables X11 graphical forwarding in interactive jobs
--pty	srun -n 1 --pty /bin/bash	Submits an interactive job that will create a terminal when the job starts.
--exclusive	--exclusive	Puts the host/s running your job into exclusive execution mode.

جامعـــة خليفـــة
Khalifa University

Thank You

ku.ac.ae