



Leibniz Supercomputing Centre  
of the Bavarian Academy of Sciences and Humanities

The background of the slide is a photograph of a modern, multi-story building, likely the LRZ facility, with a blue color overlay. The building has a mix of glass and solid panels, and there are trees in the foreground.

# A Closer Look at the LRZ Linux Cluster

2022-04-22 | J. Albert-von der Gönna



# Course Information

- The aim of this course is to provide a closer look at the characteristics of and the basic interaction with the LRZ Linux Cluster as part of the High-Performance Computing (HPC) infrastructure of the Leibniz Supercomputing Centre (LRZ)
- You will probably benefit the most if you're not yet familiar with the LRZ Linux Cluster, but plan to work on this system in the future

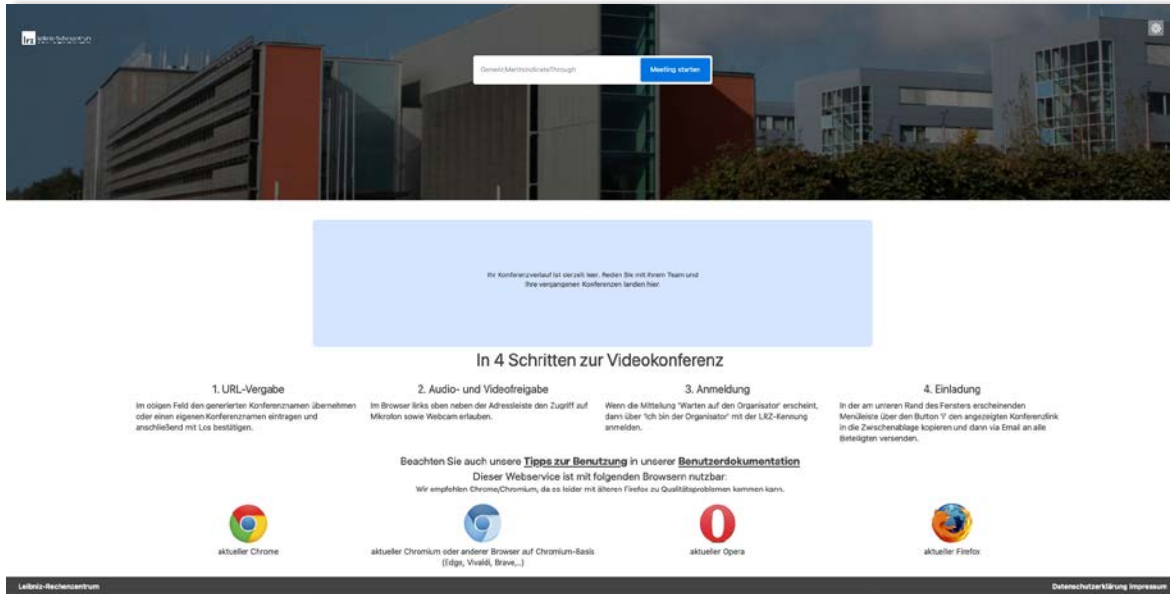
-> by the end of this workshop, you should have the basic skills to successfully interact remotely with the LRZ Linux Cluster



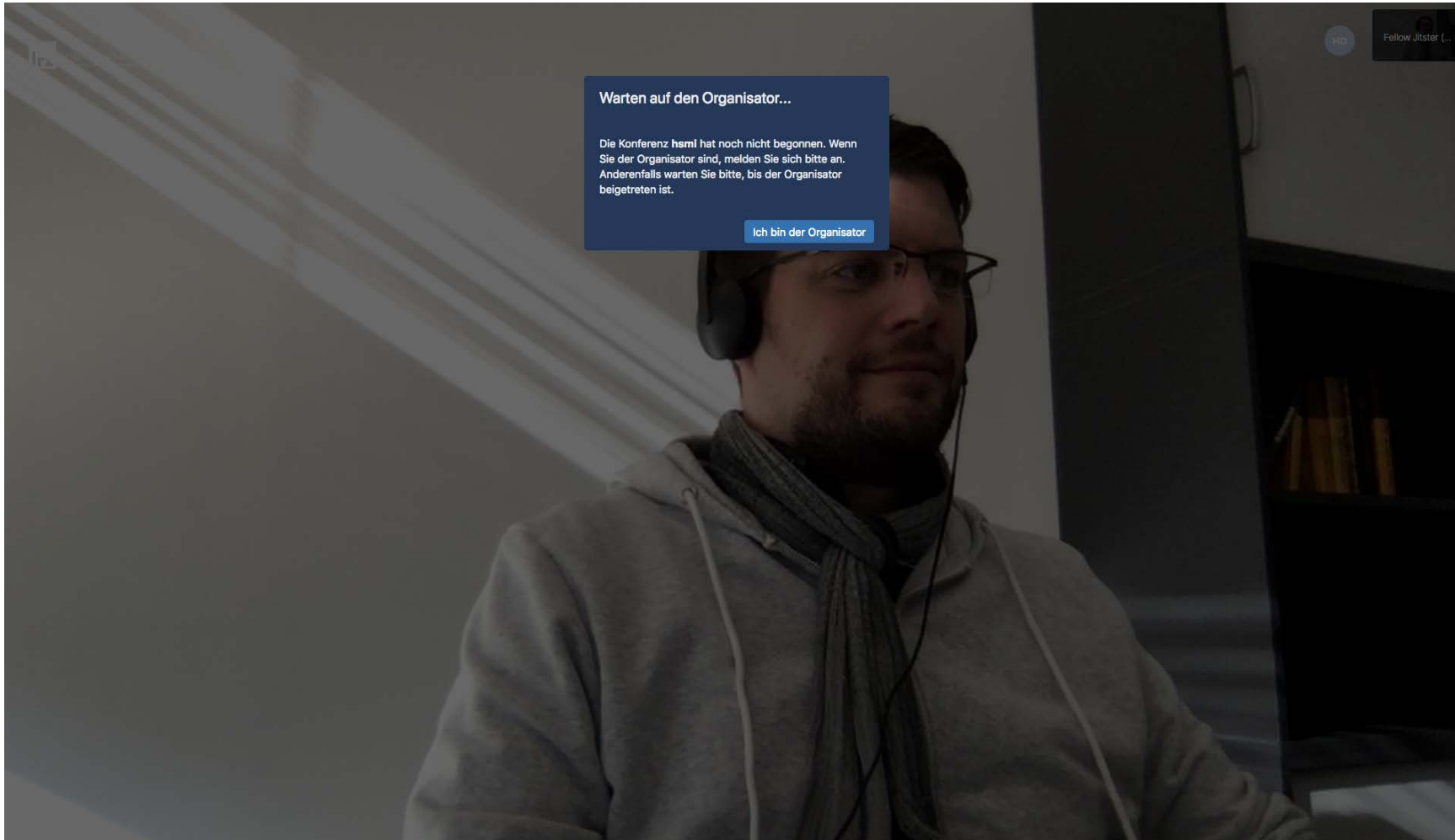
# Side Note: Collaboration Tools



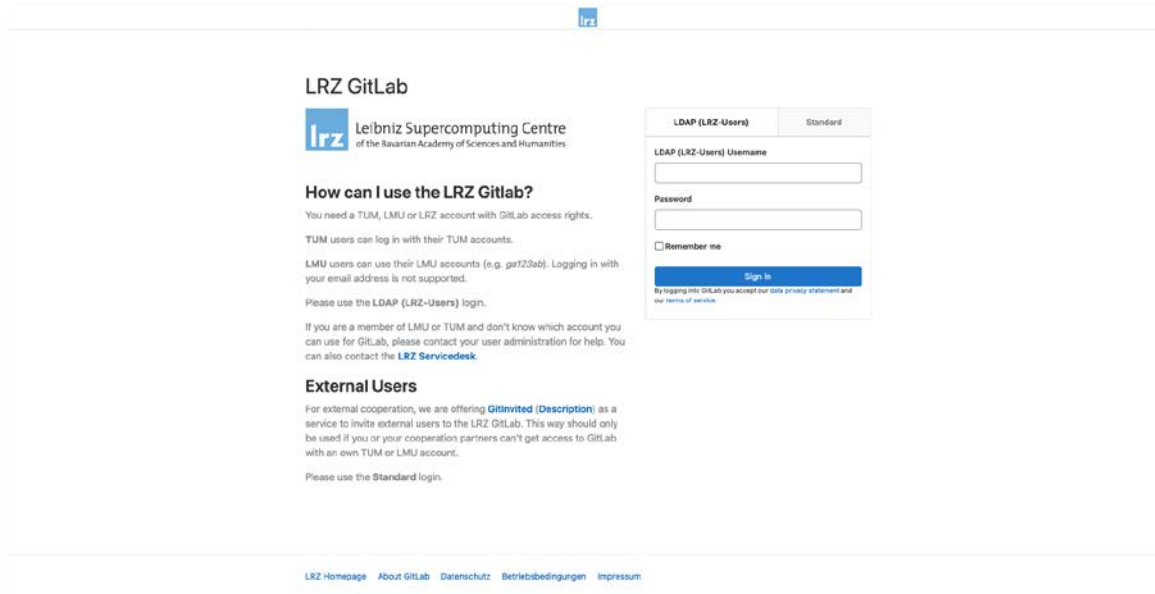
- Videoconferencing  
<https://meet.lrz.de>
- Git-repository management, issue tracker, wiki...  
<https://gitlab.lrz.de>



- Use a modern browser (Safari will likely not work properly)



First person to join has to sign in using a valid LRZ account



- Git repository manager providing wiki, issue-tracking and continuous integration and deployment pipeline features
- Every LRZ project member can invite up to 20 external users to the service (GitInvited), this is great for collaborations!



# Git Repository Management



GitLab Community Edition

Overview

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Charts

Locked Files

Issues 9,023

Merge Requests 462

CI / CD

Wiki

Snippets

Members

Collapse sidebar

GitLab.org > GitLab Community E... > Repository

master gitlab-ce / +

History Find file

Merge branch 'feature/migrate-repository-diff' into 'master' ...  
Douwe Maan committed 49 minutes ago

Name	Last commit	Last Update
.github	Address feedback about wording.	a year ago
.gitlab	Added a template for database changes	a month ago
app	Merge branch '38234-reserve-refs-replace' ...	about an hour ago
bin	Merge branch '21949-add-type-to-changelo...	a month ago
builds	Add missing builds/ folder to fix backup tests	2 years ago
changelogs	Merge branch 'ee-add-project-repository-st...	about an hour ago
config	Resolve "Better SVG Usage in the Frontend"	about 3 hours ago
db	Index projects on repository storage	a day ago
doc	Merge branch '38016-fix-bad-link-on-gitlab...	about 3 hours ago
docker	Common Docker Documentation Location in ...	3 weeks ago
features	Merge branch 'replace_project_shortcuts.fea...	about an hour ago
fixtures/emojis	Added emoji description title to award emoji ...	4 months ago
generator_templates	Add remove_concurrent_index to database h...	5 months ago
lib	Merge branch 'feature/migrate-repository-di...	49 minutes ago

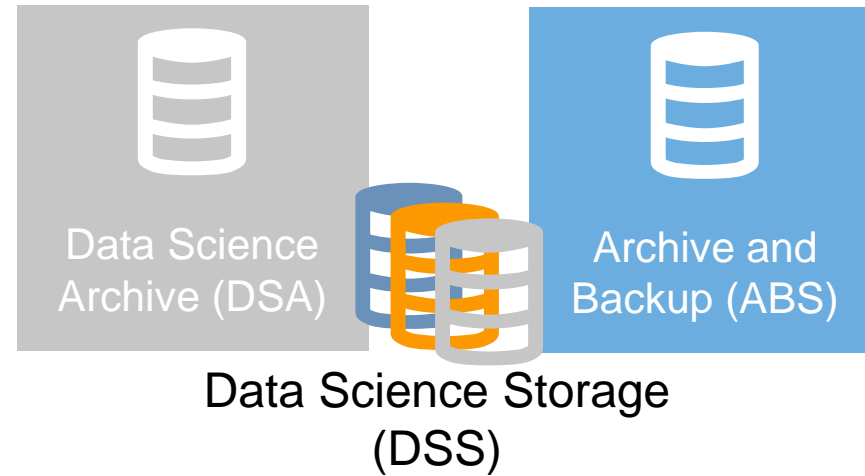
# Git Repository Management



The screenshot displays the GitLab Issue Boards interface for the 'GitLab Community Edition' project. The top navigation bar includes 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. The left sidebar shows navigation options: Overview, Repository, Issues (8,903), Merge Requests (455), CI / CD, Wiki, Snippets, and Members. The main content area is titled 'Issue Boards' and shows a 'Development' board with a search bar and 'Add list' and 'Add issues' buttons. Three columns of issues are visible:

- Backlog (3427):**
  - Only first line of pre-receive and post-receive hooks error is captured #25214 (bug)
  - Error message shows up at the wrong place in repository settings. #24319 (bug, reproduced on GitLab.com, settings)
  - Retrying a failed build repeats the Slack notification about a previous failure #21919 (bug, external services)
  - registry: deleting image tag deletes all tags with same image id #21405 (CI/CD, bug, container registry, customer)
  - Very first commit to default branch didn't close referenced issue #20930 (Documentation, bug, docs-priority, repository)
  - Sticky runners #29447 (CI/CD, Stretch, backend, docs-missing, runner)
  - Improve non-triggered manual action job detail name #22643
- UX (1129):**
  - Do a better job of communicating when MR is blocked by a locked file. #29419 (UX, bug, frontend, merge requests)
  - Unable to see user to add him to repositories #29371 (Platform, UX, bug, frontend, reproduced on GitLab.com, user management)
  - No feedback when project limit is reached #28764 (UX, bug, frontend)
  - When "No one" is allowed to push, the manual merge hint should not be shown #28171 (UX, bug, merge requests)
  - URLs in the Repository/Tags section are not blue-coloured like in Project or Wiki sections #27997 (UX, bug, repository)
  - "Project will be deleted" looks like an error occurred #26956
- frontend (1595):**
  - Import project by URL form error hides the field #28349 (bug, frontend)
  - Contribution calendar label is cut off #27839 (Accepting Merge Requests, UI polish, bug, frontend, user profile)
  - The buttons to resolve a discussion are malformed on Firefox under my Debian Stretch #26522 (bug, frontend, merge requests)
  - Do a better job of communicating when MR is blocked by a locked file. #29419 (UX, bug, frontend, merge requests)
  - Unable to see user to add him to repositories #29371 (Platform, UX, bug, frontend, reproduced on GitLab.com, user management)
  - No feedback when project limit is reached #28764 (UX, bug, frontend)

# HPC & BDAI Systems for Bavarian Universities



## LRZ Linux Cluster

CoolMUC-2    Teramem    CoolMUC-3

[lxlogin\[1-4\].lrz.de](https://lxlogin[1-4].lrz.de)  
[lxlogin8.lrz.de](https://lxlogin8.lrz.de)

## LRZ AI Systems

- “Big Data” CPU nodes
- HPE P100 node
- V100 nodes
- DGX-1 P100, DGX-1 V100
- (Multiple DGX A100)

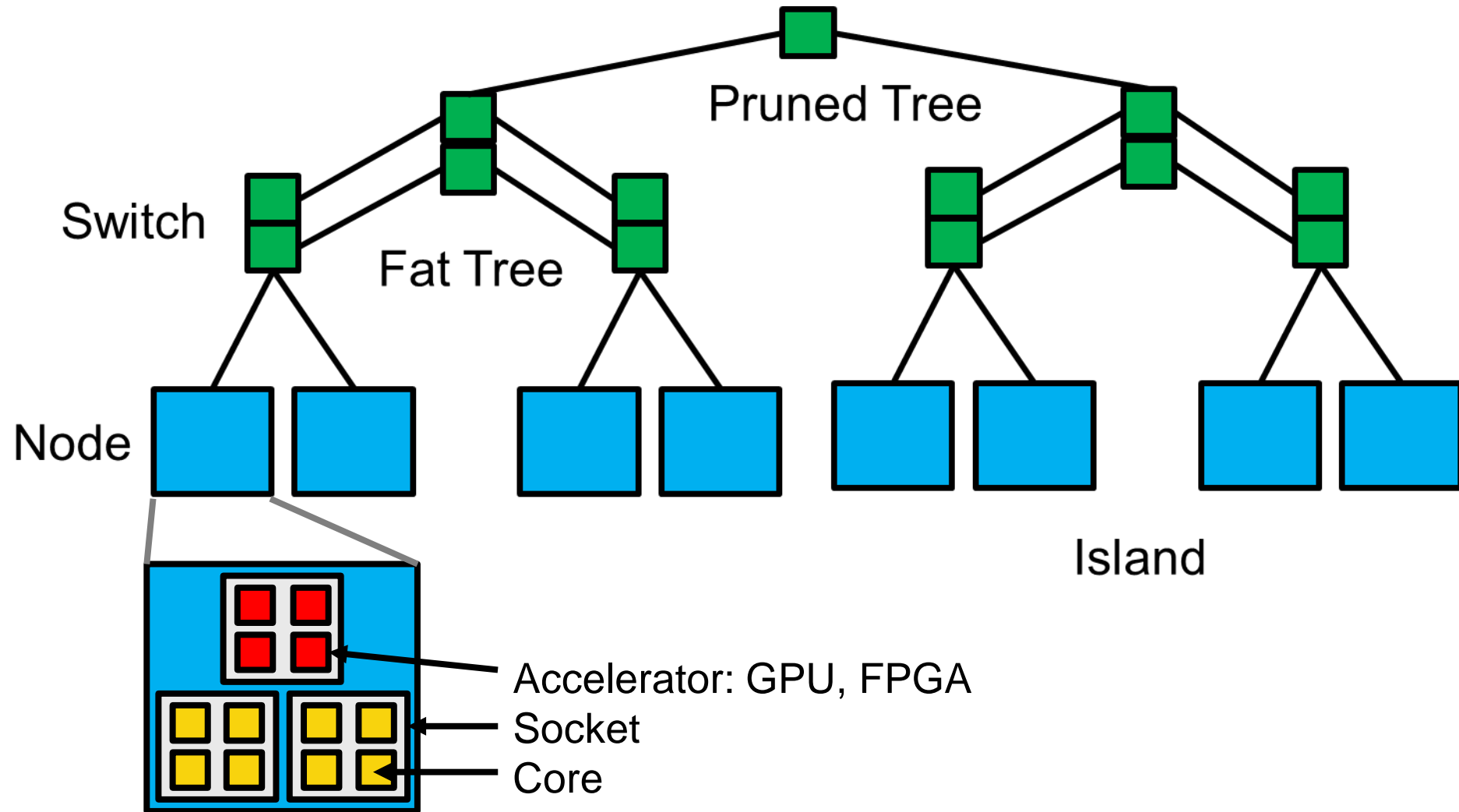
[datalab2.srv.lrz.de](https://datalab2.srv.lrz.de)  
<https://datalab3.srv.lrz.de>



## LRZ Compute Cloud

LRZ Compute Cloud  
(w/ some GPUs)

<https://cc.lrz.de>



# Linux Cluster: Hardware Overview



Name	CPU	Cores/Node	RAM/Node (GB)	Nodes (total)	Cores (total)
CoolMUC-2	Intel Xeon E5-2690 v3 ("Haswell")	28	64	812	22736
CoolMUC-3	Intel Xeon Phi ("Knights Landing")	64	96	148	9472
Teramem	Intel Xeon E7-8890 v4 ("Broadwell")	96	6144	1	96



## Linux Cluster: Access in Case LRZ Project Exists

- The master user has to check if the LRZ project is already eligible for Linux Cluster usage.
  - If not, the master user must contact the LRZ contact person for the project (advisor). The LRZ advisor will then explain the next steps to the master user.
  - If the LRZ project is eligible for Linux Cluster usage, the master user can create a new personal LRZ user ID with Linux Cluster access rights for you through the LRZ Identity Management (IDM) Portal.

The master user will need your nationality. Please provide this information. It is a necessary requirement for the export control regulations affecting all HPC/HPDA/HPAI services at LRZ.
- After you get the new user ID from your master user, please use the password reset function of the LRZ IDM Portal using your new ID and your contact e-mail address:  
<https://idmportal.lrz.de/pwreset>



## Linux Cluster: Access in Case No LRZ Project Exists

- Your chair/research group has to apply for a new LRZ project.
- Use PDF application form "Antrag auf ein LRZ-Projekt" to be found here: <https://doku.lrz.de/x/CgCiAQ> (only available in German, unfortunately)
  - Pay attention to "Gewünschte LRZ-Serviceklassen" in the application form. For Linux Cluster access you need to
    - select "High Performance Computing" and
    - fill in the phrase "Linux Cluster" at "andere Dienste:"
  - Send the filled in and signed application form to the responsible LRZ contact person (advisor). The original document is needed (you may send a scanned copy to speed up the process, but this does not replace sending the physical letter via snail mail).
- The master user of the newly requested LRZ project will get instructions from the LRZ advisor:
  - Fill out the Service Request Template for "Linux Cluster Project Activation" and submit it to LRZ Servicedesk (<https://servicedesk.lrz.de/en/ql/createsr/12>)
  - Afterwards, the Linux Cluster access for the new LRZ project is typically approved
  - Now, your new master user can create new LRZ user IDs with access to the Linux Cluster (see previous slide)



# Data Storage: Linux Cluster



- DSS-backed home directory `$HOME` (managed by LRZ)
  - 100GB per user
  - Access: `/dss/dsshhome1/lxc###/<user>`
  - Automatic tape backup and file system snapshots (see `"/dss/dsshhome1/.snapshots/"` directory)
  - All your important files/anything you invested a lot of work into should be here





# Data Storage: Linux Cluster



- DSS project storage
  - Up to 10 TB per project upon request, shared among project members
  - Access: `$ dssusrinfo all`
  - Configuration (e.g. exports, backup, quota) to be managed by data curator
  - Use this for e.g. large raw data (and consider backup options)



## Data Storage: Linux Cluster

- Legacy `$SCRATCH` (scratch file system, “temporary file system”)
  - 1.4 PB, shared among all users
  - Access: `/gpfs/scratch/<group>/<user>`
- New `$SCRATCH_DSS` (not yet available on CoolMUC-2 compute nodes)
  - 3.1 PB, shared among all users
  - Access: `/dss/lxclscratch/##/<user>`
- No backup (!) and sliding window file deletion, i.e. old files will eventually be deleted (!!)  
– a data retention time of approx. 30 days may be assumed, but is not guaranteed
- This is the place for e.g. very large, temporary files or intermediate results, directly feeding into additional analyses
- Data integrity is not guaranteed. Do not save any important data exclusively on these file systems! Seriously, don't do it!





- Let's get started:  
Connect to the CoolMUC-2 segment  
of the Linux Cluster
- From a terminal application:  
`$ ssh <user>@lxlogin1.lrz.de`

# Linux Cluster – CoolMUC-2



```
user@localhost:~$ ssh user@lxlogin1.lrz.de
```

- For CoolMUC-2 you can use the login nodes  
lxlogin1.lrz.de or  
lxlogin2.lrz.de or  
lxlogin3.lrz.de or  
lxlogin4.lrz.de
- (see <https://doku.lrz.de/x/AAaVAg> for all cluster segments/systems)

```
user@localhost:~$ ssh user@lxlogin1.lrz.de
The authenticity of host 'lxlogin1.lrz.de (129.187.20.101)' can't be established.
ECDSA key fingerprint is SHA256:Q2NG5ofc7v/ew1kZYXcEuu69T3ESoIUkY9bITwNKJ5g.
Are you sure you want to continue connecting (yes/no)?
```

- The first time you connect to a (new) system, this message is expected, as the public-key of the remote system is not yet known to your local system, see <https://superuser.com/questions/421997/what-is-a-ssh-key-fingerprint-and-how-is-it-generated> for details.
- Information about the public keys of the LRZ Linux Cluster can be found here: <https://doku.lrz.de/x/AAaVAg>
- Type “yes” to import the public key locally and to continue.

# Linux Cluster – CoolMUC-2



```
user@localhost:~$ ssh user@lxlogin1.lrz.de
The authenticity of host 'lxlogin1.lrz.de (129.187.20.101)' can't be established.
ECDSA key fingerprint is SHA256:Q2NG5ofc7v/ew1kZYXcEuu69T3ESoIUkY9bITwNKJ5g.
Are you sure you want to continue connecting (yes/no)?
Warning: Permanently added 'lxlogin1.lrz.de' (ECDSA) to the list of known hosts.
Password:
```

- You can now continue by typing your password.  
As this may become a repetitive burden, you may choose to do the following instead...
- (Interrupt the password prompt by pressing `Ctrl+C`)

- Using a local terminal, add your SSH public key (not the private one!) to the authorized keys on the Linux Cluster!
- On Linux, this is the content of `~/.ssh/<your_key>.pub` on your local machine...
- ... which should go into `~/.ssh/authorized_keys` on the login node
- Use the command `ssh-copy-id <user>@lxlogin1.lrz.de` on Linux, you may have to do it differently/manually on macOS and Windows (but only once)

```
di36pez@ubuntu1804: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
di36pez@ubuntu1804:~$ ssh di36pez@lxlogin5.lrz.de
The authenticity of host 'lxlogin5.lrz.de (129.187.20.105)' can't be established
.
ECDSA key fingerprint is SHA256:YmTuVciNdQzoZXpiDC4encMuUa8WIjJuA4NqmXaXgeM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lxlogin5.lrz.de,129.187.20.105' (ECDSA) to the list
of known hosts.
Password:

di36pez@ubuntu1804:~$ ssh-copy-id di36pez@lxlogin5.lrz.de
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'di36pez@lxlogin5.lrz.de'"
and check to make sure that only the key(s) you wanted were added.

di36pez@ubuntu1804:~$ █
```

Demo content –  
use the current  
login nodes  
instead!



- Connect to the cluster again, it should now work without password!  
Crikey!
- You can use the `logout` or `exit` commands to close the connection

# Linux Cluster – CoolMUC-2



```
user@localhost:~$ ssh user@lxlogin1.lrz.de
Last login: Fri Apr 1 11:11:11 2022 from SOME.IP.ADDRESS
Welcome to the CoolMUC2 Infiniband cluster, one of the Linux cluster systems
operated by Leibniz Supercomputing Centre (LRZ).

Please do not run any extensive computational programs on login nodes.
Instead, please submit SLURM batch scripts for production jobs, and SLURM
interactive shells for testing and short-running programs.
Misuse of the interactive resources will lead to violating accounts being
blocked from access to the cluster.
!!! Please note in particular this pertains to specifying invalid !!!
!!! mail addresses in SLURM scripts, please read             !!!
https://doku.lrz.de/display/PUBLIC/Available+SLURM+clusters+and+features
-----
Documentation:  https://doku.lrz.de/display/PUBLIC/Linux+Cluster
Messages/System Status:
                 https://doku.lrz.de/display/PUBLIC/High+Performance+Computing
-----
Mar 18, 2022: Please note the announcement for a scheduled maintenance
of all cluster systems at https://www.lrz.de/aktuell/ali00936.html
-----
spack/release/22.2.1 22.2.1 release linux-sles15
intel-mpi: using intel wrappers for mpicc, mpif77, etc

user@cm2login1:~$
```

```
user@localhost:~$ ssh user@lxlogin1.lrz.de
Last login: Fri Apr 1 11:11:11 2022 from SOME.IP.ADDRESS
Welcome to the CoolMUC2 Infiniband cluster, one of the Linux cluster
systems operated by Leibniz Supercomputing Centre (LRZ).
Please do not run any extensive computational programs on login nodes.
Instead, please submit SLURM batch scripts for production jobs, and SLURM
interactive shells for testing and short-running programs.
Misuse of the interactive resources will lead to violating accounts being
blocked from access to the cluster.
!!! Please note in particular this pertains to specifying invalid !!!
!!! mail addresses in SLURM scripts, please read             !!!
https://doku.lrz.de/display/PUBLIC/Available+SLURM+clusters+and+features
-----
Documentation: https://doku.lrz.de/display/PUBLIC/Linux+Cluster
Messages/System Status:
https://doku.lrz.de/display/PUBLIC/High+Performance+Computing
-----
Mar 18, 2022: Please note the announcement for a scheduled maintenance
of all cluster systems at https://www.lrz.de/aktuell/ali00936.html
-----
spack/release/22.2.1 22.2.1 release linux-sles15
intel-mpi: using intel wrappers for mpicc, mpif77, etc

user@cm2login1:~$
```

- This is the message of the day provided by the system administrators
- Take note of it, it may contain important information about the system status, scheduled maintenances, etc. ...

- Get your bearings... where did you end up on the file system?  
Use \$ `pwd` to print your current directory.

- Use `dssusrinfo all` to query DSS-containers available to you (for additional options, see `dssusrinfo -h`)
- Look at and explore `$HOME` (this is an environment variable):  
`$ echo $HOME`

- Modules allow for the dynamic modification of environment variables, e.g. they provide a flexible way to access various applications and libraries available on the system
- List the currently active modules (loaded by default):  
`$ module list`

# Environment Modules



```
user@cm2login1:~$ module list
Currently Loaded Modulefiles:
  1) admin/1.0    2) tempdir/1.0  3) lrz/1.0    4) spack/22.2.1
  5) intel-oneapi-compilers/2021.4.0  6) intel-mkl/2020  7) intel-mpi/2019-intel
user@cm2login1:~$
```

Get more info about e.g. the lrz module:

```
$ module show lrz
```

# Environment Modules



```
user@cm2login1:~$ module show lrz
```

```
-----  
/lrz/sys/share/modules/files_sles15/environment/lrz/1.0:
```

```
module-whatism  {Environment: Default setup for all LRZ users}  
setenv          LRZ_ARCH x86_64_intel  
append-path    PATH /lrz/sys/tools/slurm_utils/bin  
setenv          SALLOC_PARTITION cm2_inter  
setenv          LRZ_OS {SUSE Linux Enterprise Server 15 SP1}  
setenv          LRZ_OS_VER 15  
setenv          LRZ_OS_SUBVER 1  
setenv          LRZ_NOCHECK yes  
setenv          INTEL_LICENSE_FILE /lrz/sys/intel/licenses  
setenv          LRZ_LICENSE {open source - no access restrictions}
```



- Suppose you need to use MATLAB
- It is not generally available (try `$ which matlab`)
- ... or is it?

Search for available modules:

```
$ module available matlab or
```

```
$ module av matlab
```

# Environment Modules



```
user@cm2login1:~$ which matlab
which: no matlab in (/lrz/sys/tools/intel-mpi-wrappers/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-
mpi/2019.12.320-gcc-
wx7cjl原因/compilers_and_libraries_2020.4.320/linux/mpi/intel64/libfabric/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x
86_64/intel-mpi/2019.12.320-gcc-
wx7cjl原因/compilers_and_libraries_2020.4.320/linux/mpi/intel64/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/inte
l-mpi/2019.12.320-gcc-wx7cjl原因/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-mkl/2020.4.304-gcc-
cmdxw76/mkl/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-mkl/2020.4.304-gcc-
cmdxw76/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-oneapi-compilers/2021.4.0-gcc-
xrzccgg/debugger/10.2.4/gdb/intel64/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-oneapi-
compilers/2021.4.0-gcc-xrzccgg/compiler/2021.4.0/linux/lib/oclfpga/llvm/aocl-
bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-oneapi-compilers/2021.4.0-gcc-
xrzccgg/compiler/2021.4.0/linux/lib/oclfpga/bin:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-oneapi-
compilers/2021.4.0-gcc-
xrzccgg/compiler/2021.4.0/linux/bin/intel64:/dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/x86_64/intel-oneapi-
compilers/2021.4.0-gcc-
xrzccgg/compiler/2021.4.0/linux/bin:/lrz/sys/tools/modules/4.6.1/bin:/lrz/sys/bin:/dss/dsshome1/06/di36pez/bin:/usr/local/b
in:/usr/bin:/bin:/lrz/sys/tools/slurm_utils/bin)
user@cm2login1:~$ module av matlab
----- /lrz/sys/spack/release/22.2.1/modules/x86_64/linux-sles15-x86_64 -----
matlab-mcr/R2020a_Update5  matlab-mcr/R2021a_Update5  matlab-mcr/R2021b_Update2      matlab/R2021b_Update3-generic
matlab-mcr/R2020b_Update3  matlab-mcr/R2021a_Update6  matlab-mcr/R2022a              matlab/R2022a-generic
matlab-mcr/R2021a          matlab-mcr/R2021b          matlab/R2021a_Update6-generic

----- /lrz/sys/share/modules/files_sles15/applications -----
matlab-inter/coolmuc-2  matlab-inter/coolmuc-3
```

- Look at all these options...!
- Most modules are maintained using the Spack package manager, i.e. typically prioritize modules in the “/lrz/sys/spack/...” path!
- Load any module you like, e.g. the latest MATLAB version:  
`$ module load matlab/R2022a-generic`

# Environment Modules



```
user@cm2login1:~$ module load matlab/R2022a-generic
```

WARNING: Please note that the dynamic loader is overloaded by this MATLAB environment module!

Please note further that the setting of the KMP\_AFFINITY environment variable is also modified by MATLAB environment module! This may have negative impact on the performance and functionality of other OpenMP based programs.

Use a different shell to start programs other than MATLAB, otherwise those programs may not function properly.

```
user@cm2login1:~$ which matlab  
/dss/dsshhome1/lrz/sys/spack/release/22.2.1/opt/x86_64/matlab/R2022a-gcc-6wxszwk/bin/matlab
```

- Modules can/should be unloaded when you don't need them anymore (e.g. before trying another application version):  
`$ module unload <module/version>`
- Loading modules is not persistent across sessions, i.e. once you log out and back in again, only the default modules will be loaded!
- For further documentation, see <https://modules.readthedocs.io/en/latest/module.html>



- Slurm is a job scheduler:
  - Allocates access to resources (time, memory, nodes/cores)
  - Provides framework for starting, executing, and monitoring work
  - Manages queue of pending jobs (enforcing “fair share” policy)
- Use the `sinfo` command to get information about the available clusters

```
$ sinfo --clusters=all or, shortened:
```

```
$ sinfo -M all
```



```
user@cm2login1:~$ sinfo -M all -s
```

```
CLUSTER: biohpc_gen
```

PARTITION	AVAIL	TIMELIMIT	NODES(A/I/O/T)	NODELIST
biohpc_gen_highmem	up	21-00:00:0	3/10/0/13	hleg1409n[01-13]
biohpc_gen_production	up	14-00:00:0	3/10/0/13	hleg1409n[01-13]
biohpc_gen_normal	up	2-00:00:00	3/10/0/13	hleg1409n[01-13]
biohpc_gen_inter*	up	12:00:00	3/10/0/13	hleg1409n[01-13]

```
CLUSTER: c2pap
```

PARTITION	AVAIL	TIMELIMIT	NODES(A/I/O/T)	NODELIST
c2pap_parallel	up	2-00:00:00	114/6/0/120	i23r07c01s[01-12],i23r07c02s[01-12],i23r07c03s[01-12],i23r07c04s[01-12],i23r07c05s[01-12],i23r08c01s[01-12],i23r08c02s[01-12],i23r08c03s[01-12],i23r08c04s[01-12],i23r08c05s[01-12]
c2pap_serial*	up	2-00:00:00	114/6/0/120	i23r07c01s[01-12],i23r07c02s[01-12],i23r07c03s[01-12],i23r07c04s[01-12],i23r07c05s[01-12],i23r08c01s[01-12],i23r08c02s[01-12],i23r08c03s[01-12],i23r08c04s[01-12],i23r08c05s[01-12]
c2pap_preempt	up	2-00:00:00	114/6/0/120	i23r07c01s[01-12],i23r07c02s[01-12],i23r07c03s[01-12],i23r07c04s[01-12],i23r07c05s[01-12],i23r08c01s[01-12],i23r08c02s[01-12],i23r08c03s[01-12],i23r08c04s[01-12],i23r08c05s[01-12]

```
CLUSTER: cm2
```

PARTITION	AVAIL	TIMELIMIT	NODES(A/I/O/T)	NODELIST
cm2_std*	up	3-00:00:00	386/11/7/404	i22r01c01s[01-12],i22r01c02s[01-12],i22r01c03s[01-12],i22r01c04s[01-12],i22r01c05s[01-12],i22r01c06s[01-12],i22r02c01s[01-12],i22r02c02s[01-12],i22r02c03s[01-12],i22r02c04s[01-12],i22r02c05s[01-12],i22r02c06s[01-12],i22r03c01s[01-12],i22r03c02s[01-12],i22r03c03s[01-12],i22r03c04s[01-12],i22r03c05s[01-12],i22r03c06s[01-12],i22r04c01s[01-12],i22r04c02s[01-12],i22r04c03s[01-12],i22r04c04s[01-12],i22r04c05s[01-12],i22r04c06s[01-12],i22r05c01s[01-12],i22r05c02s[01-12],i22r05c03s[01-12],i22r05c04s[01-12],i22r05c05s[01-12],i22r06c01s[01-12],i22r06c02s[01-12],i22r06c03s[01-12],i22r06c04s[01-12],i22r06c05s[01-12],i22r07c02s[01-12],i22r07c04s[07-12]

```
...
```

- Look for the cluster segments
  - inter (allows for interactive usage)
  - cm2 (the main CoolMUC-2 cluster)
  - serial (shared nodes for serial jobs)
- What is their current status?
- Get information about a specific cluster segment, e.g.

```
$ sinfo -M inter or
```

```
$ sinfo -M cm2
```

# CoolMUC-2 Overview



Slurm cluster	Slurm partition	Node range per Job	Slurm job settings
cm2	cm2_large	25-64	--clusters=cm2 --partition=cm2_large --qos=cm2_large
	cm2_std	3-24	--clusters=cm2 --partition=cm2_std --qos=cm2_std
cm2_tiny	cm2_tiny	1-4	--clusters=cm2_tiny
serial	serial_std	1	--clusters=serial --partition=serial_std --mem=<memory_per_node>MB
	serial_long	1	--clusters=serial --partition=serial_long --mem=<memory_per_node>MB
inter	cm2_inter	1-4	--clusters=inter --partition=cm2_inter
	teramem_inter	1	--clusters=inter --partition=teramem_inter

For additional details see <https://doku.lrz.de/display/PUBLIC/Job+Processing+on+the+Linux-Cluster>



```
usercm2login1:~$ sinfo -M inter -s
```

```
CLUSTER: inter
```

PARTITION	AVAIL	TIMELIMIT	NODES(A/I/O/T)	NODELIST
mpp3_inter*	up	2:00:00	0/3/0/3	mpp3r03c05s[01-03]
teramem_inter	up	10-00:00:0	1/0/0/1	teramem1
cm2_inter	up	2:00:00	12/0/0/12	i22r07c05s[01-12]
cm2_inter_large_mem	up	4-00:00:00	0/6/0/6	i22r07c01s[01-06]

```
user@cm2login1:~$ sinfo -M cm2 -s
```

```
CLUSTER: cm2
```

PARTITION	AVAIL	TIMELIMIT	NODES(A/I/O/T)	NODELIST
cm2_std*	up	3-00:00:00	377/20/7/404	i22r01c01s[01-12],i22r01c02s[01-12],i22r01c03s[01-12],i22r01c04s[01-12],i22r01c05s[01-12],i22r01c06s[01-12],i22r02c01s[01-12],i22r02c02s[01-12],i22r02c03s[01-12],i22r02c04s[01-12],i22r02c05s[01-12],i22r02c06s[01-12],i22r03c01s[01-12],i22r03c02s[01-12],i22r03c03s[01-12],i22r03c04s[01-12],i22r03c05s[01-12],i22r03c06s[01-12],i22r04c01s[01-12],i22r04c02s[01-12],i22r04c03s[01-12],i22r04c04s[01-12],i22r04c05s[01-12],i22r05c01s[01-12],i22r05c02s[01-12],i22r05c03s[01-12],i22r05c04s[01-12],i22r05c05s[01-12],i22r06c01s[01-12],i22r06c02s[01-12],i22r06c03s[01-12],i22r06c04s[01-12],i22r06c05s[01-12],i22r07c02s[11-12],i22r07c04s[07-12]
cm2_large	up	2-00:00:00	377/20/7/404	i22r01c01s[01-12],i22r01c02s[01-12],i22r01c03s[01-12],i22r01c04s[01-12],i22r01c05s[01-12],i22r01c06s[01-12],i22r02c01s[01-12],i22r02c02s[01-12],i22r02c03s[01-12],i22r02c04s[01-12],i22r02c05s[01-12],i22r02c06s[01-12],i22r03c01s[01-12],i22r03c02s[01-12],i22r03c03s[01-12],i22r03c04s[01-12],i22r03c05s[01-12],i22r03c06s[01-12],i22r04c01s[01-12],i22r04c02s[01-12],i22r04c03s[01-12],i22r04c04s[01-12],i22r04c05s[01-12],i22r05c01s[01-12],i22r05c02s[01-12],i22r05c03s[01-12],i22r05c04s[01-12],i22r05c05s[01-12],i22r06c01s[01-12],i22r06c02s[01-12],i22r06c03s[01-12],i22r06c04s[01-12],i22r06c05s[01-12],i22r07c02s[11-12],i22r07c04s[07-12]

```
user@cm2login1:~$
```

- The inter cluster can be used for interactive resource allocation:  
`$ salloc -p cm2_inter -N 1`

# Interactive Allocation



```
user@cm2login1:~$ salloc -p cm2_inter -N 1
salloc: Granted job allocation 141265
user@i22r07c05s06:~$ hostname
i22r07c05s06
user@i22r07c05s06:~$ exit
exit
salloc: Relinquishing job allocation 141265
user@cm2login1:~$
```

- Notice the change of the hostname, you're now logged in on a compute node!
- See <https://doku.lrz.de/display/PUBLIC/Running+parallel+jobs+on+the+Linux-Cluster#RunningparalleljobsontheLinuxCluster-InteractiveSLURMshellforparalleltesting> and <https://doku.lrz.de/x/MgKoAg> for further details
- For production jobs, you want to prepare and submit batch scripts – they tell Slurm about the resources you need and the scripts/programs you want to run

```
#!/bin/bash
#SBATCH --clusters=cm2_tiny
#SBATCH --nodes=1

module load slurm_setup

./<executable>
```

- A very minimal example of a job script (not recommended, but working in principle), requesting
  - a single, exclusive node (with 28 cores)
  - of the cm2\_tiny partition/cluster, part of
  - the CoolMUC-2 system
- Submit this job script to the queue:  
\$ sbatch <script.sh>

# Job Processing – Give it a try...



```
#!/bin/bash
#SBATCH --clusters=serial
#SBATCH --partition=serial_std

module load slurm_setup

hostname
```

- Create a new folder in your home directory (e.g. “playground”) and change into it.
- Create this very, very minimal example of a job script (again, this is generally not recommended!) and save it as “script.sh”. What will it do?
- Submit this job script to the queue:  
`$ sbatch script.sh`
- Keep your eyes open for output in the current folder. What can you find?

```
#!/bin/bash
#SBATCH -J <job_name>
#SBATCH -o ./%x.%j.%N.out
#SBATCH -D ./
#SBATCH --get-user-env
#SBATCH --clusters=cm2
#SBATCH --partition=cm2_std
#SBATCH --nodes=3
#SBATCH --ntasks-per-node=28
#SBATCH --mail-type=end
#SBATCH --mail-user=<email_address>@<domain>
#SBATCH --export=NONE
#SBATCH --time=08:00:00

module load slurm_setup

mpiexec -n $SLURM_NTASKS ./<executable>
```

- A more practical example...
  - assigning a job name
  - defining custom output file(s)
  - setting a working directory
  - configuring mail notifications
  - managing the environment
  - limiting walltime explicitly
- See documentation for more details:  
<https://doku.lrz.de/x/AgaVAg>

- Use the `squeue` command to query information about your jobs in the Slurm scheduling queue, e.g. of the cm2 cluster:  
`$ squeue -M cm2 -u <user>`
- If you're interested in the approx. start time of your pending jobs (in the the cm2 queue):  
`$ squeue -M cm2 -u <user> --start`
- Display accounting data of (finished) jobs by use of the `sacct` command, e.g.  
`$ sacct -M cm2 -u <user>`
- Per default, this is limited to today's jobs, add the `-S` option to specify a user-defined date:  
`$ sacct ... -S <YYYY-MM-DD>`