Hands-on AI based 3D Vision
Summer Semester 25

Tutorial – TCML Cluster

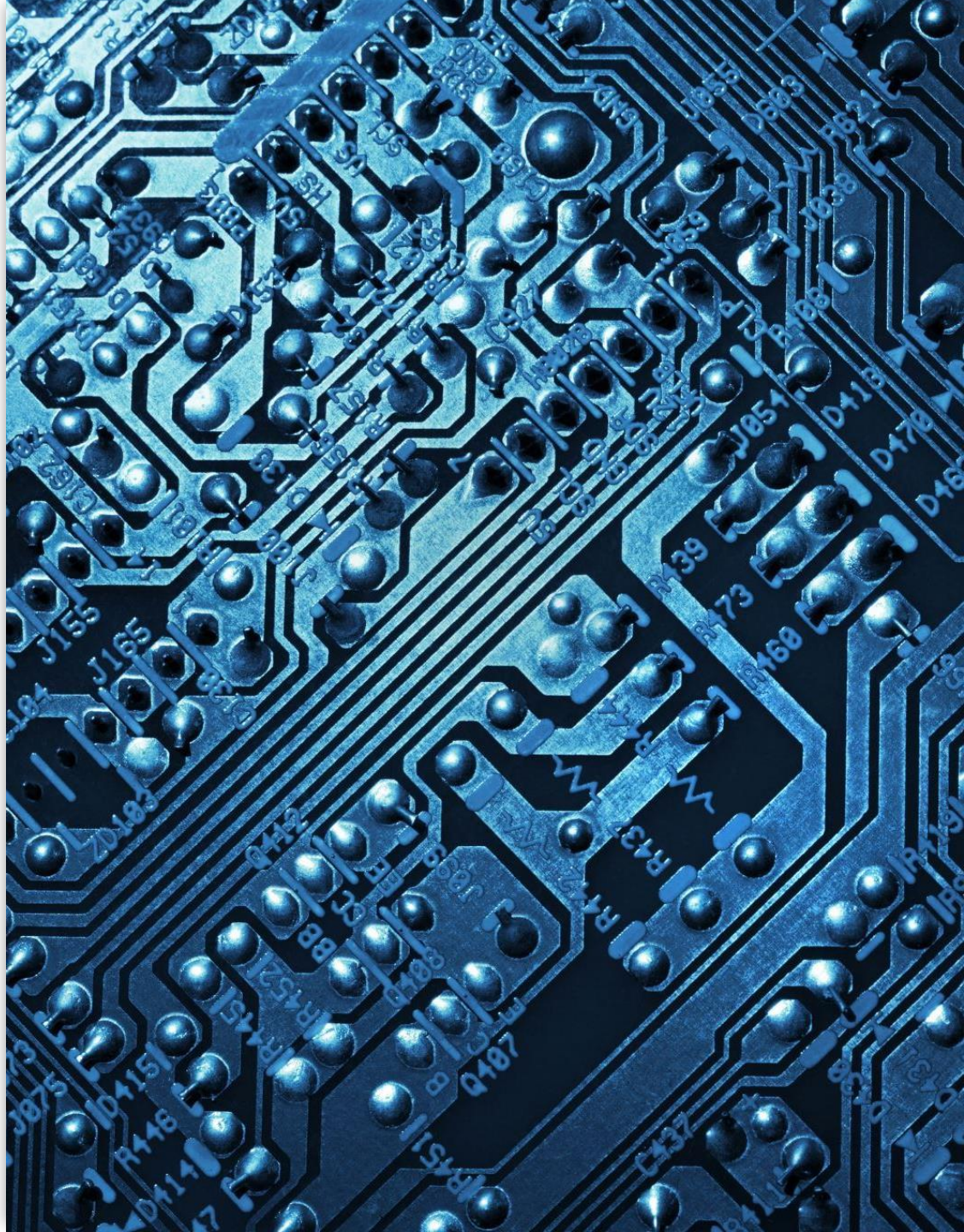
TAs:

Andrea Sanchietti, Niklas Berndt, Eyvaz Najafli

University of Tübingen / MPI-Informatics

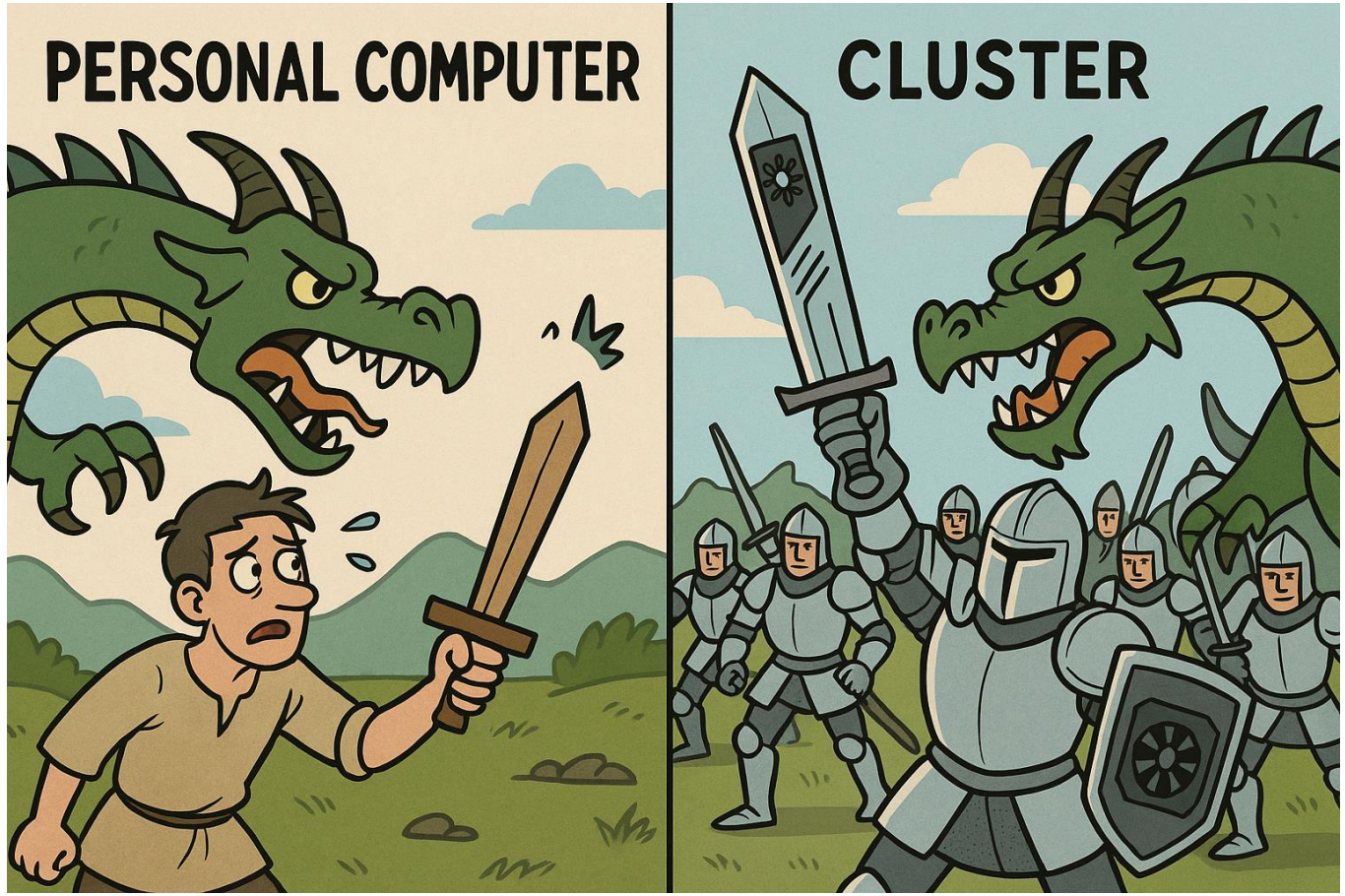
Cluster: What? How? Why? Can I eat it?

- A **cluster** is a group of powerful non-edible computers (called **nodes**) connected together to perform computations.



Cluster: What? How? Why? Can I eat it?

- Clusters are designed to handle **heavy computational tasks** and large-scale **data processing** that would be too slow or impossible on a personal computer.



Cluster: What? How? Why? Can I eat it?

- You **don't run your code directly** on the cluster like you would on your laptop.

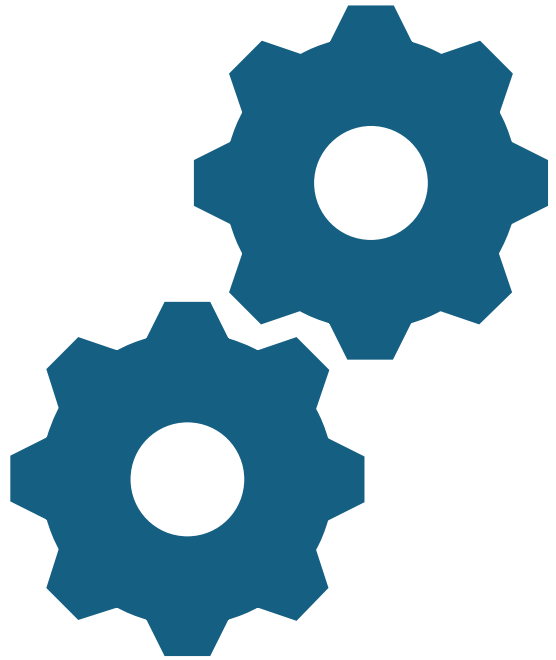
Cluster: What? How? Why? Can I eat it?

Instead, you:

- **Write a job script** – this tells the cluster:
  1. What program you want to run
  2. How many resources you need (e.g., CPUs, GPUs, memory)
  3. How long your job will take
- **Submit the job to a scheduler** – a system that organizes and queues all users' jobs.
- The **scheduler allocates resources** when available and runs your job.
- Finally, you get your results.

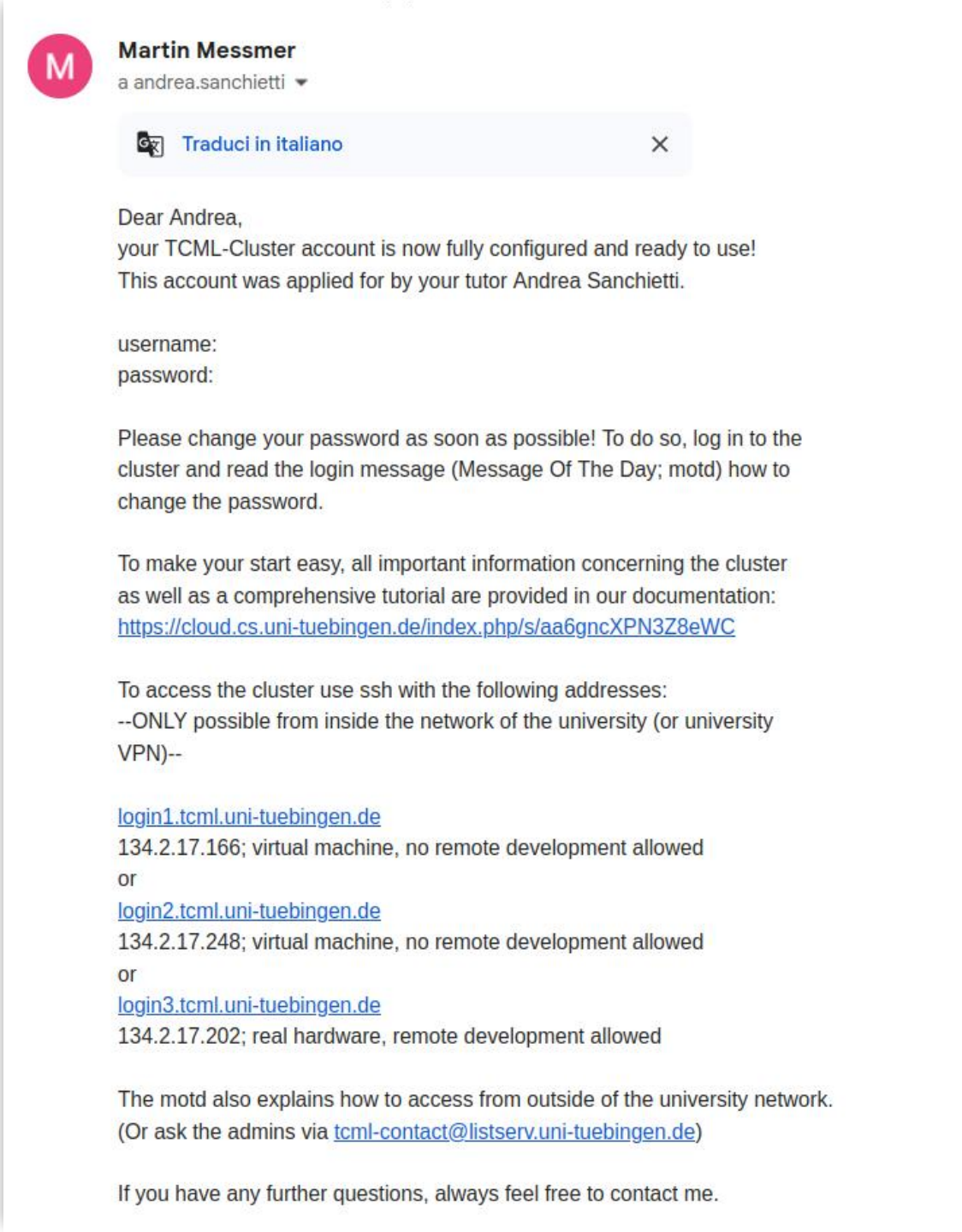
Why Use the Cluster?

- **Training deep learning models** using multiple GPUs.
- **Running simulations** or experiments that take hours or days.
- **Homeworks of** "Hands-on AI Based 3D Vision".
- ~~Mining cryptos and ddossing opponents in online matches~~



Login to the cluster

- In the past week, you should have received an email from the cluster admin



Martin Messmer
a andrea.sanchietti

Traduci in italiano

Dear Andrea,
your TCML-Cluster account is now fully configured and ready to use!
This account was applied for by your tutor Andrea Sanchietti.

username:
password:

Please change your password as soon as possible! To do so, log in to the cluster and read the login message (Message Of The Day; motd) how to change the password.

To make your start easy, all important information concerning the cluster as well as a comprehensive tutorial are provided in our documentation:
https://cloud.cs.uni-tuebingen.de/index.php/s/aa6gncXPN3Z8eWC

To access the cluster use ssh with the following addresses:
--ONLY possible from inside the network of the university (or university VPN)--

login1.tcml.uni-tuebingen.de
134.2.17.166; virtual machine, no remote development allowed
or
login2.tcml.uni-tuebingen.de
134.2.17.248; virtual machine, no remote development allowed
or
login3.tcml.uni-tuebingen.de
134.2.17.202; real hardware, remote development allowed

The motd also explains how to access from outside of the university network.
(Or ask the admins via tcml-contact@listserv.uni-tuebingen.de)

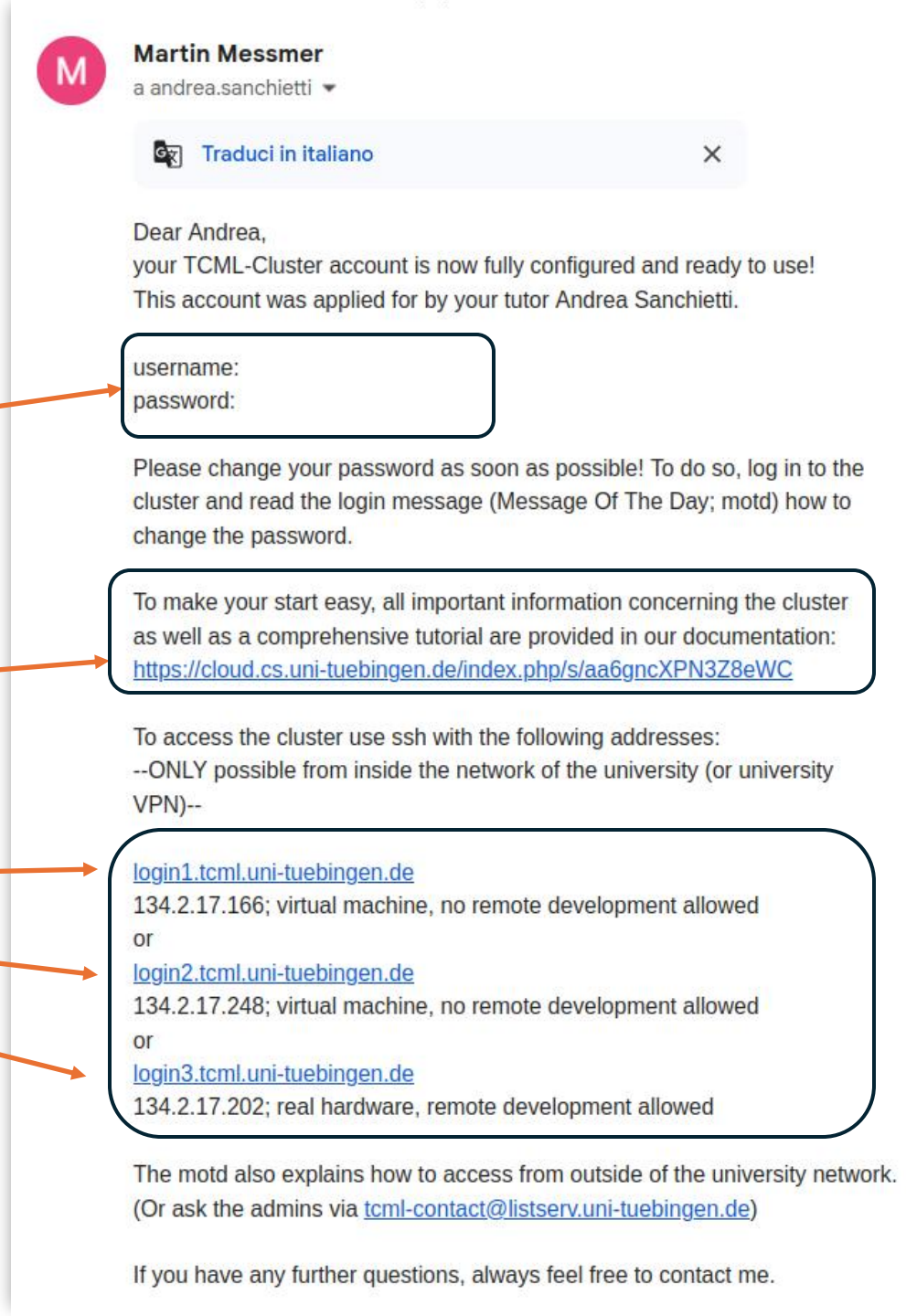
If you have any further questions, always feel free to contact me.

Login to the cluster

Login Information

Documentation and Tutorial

Login Nodes



Martin Messmer
a andrea.sanchietti

Traduci in italiano

Dear Andrea,
your TCML-Cluster account is now fully configured and ready to use!
This account was applied for by your tutor Andrea Sanchietti.

username:
password:

Please change your password as soon as possible! To do so, log in to the cluster and read the login message (Message Of The Day; motd) how to change the password.

To make your start easy, all important information concerning the cluster as well as a comprehensive tutorial are provided in our documentation:
https://cloud.cs.uni-tuebingen.de/index.php/s/aa6gncXPN3Z8eWC

To access the cluster use ssh with the following addresses:
--ONLY possible from inside the network of the university (or university VPN)--

login1.tcml.uni-tuebingen.de
134.2.17.166; virtual machine, no remote development allowed
or
login2.tcml.uni-tuebingen.de
134.2.17.248; virtual machine, no remote development allowed
or
login3.tcml.uni-tuebingen.de
134.2.17.202; real hardware, remote development allowed

The motd also explains how to access from outside of the university network.
(Or ask the admins via tcml-contact@listserv.uni-tuebingen.de)

If you have any further questions, always feel free to contact me.

Login to the cluster

Use login nodes to log in the cluster console

```
ssh stud201@login1.tcml.uni-tuebingen.de
```

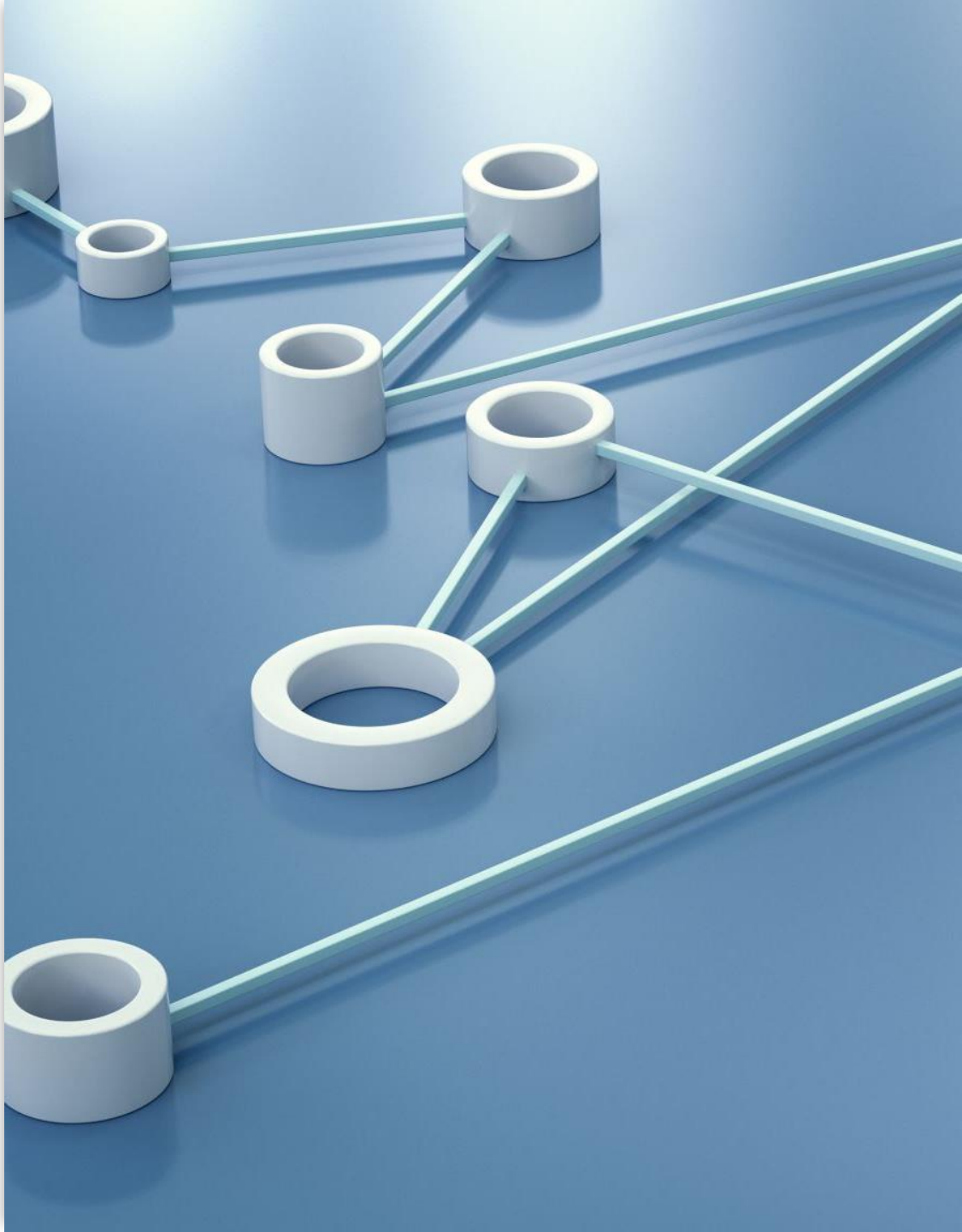
```
andrea@andrea-workstation ~> ssh stud201@login1.tcml.uni-tuebingen.de
The authenticity of host 'login1.tcml.uni-tuebingen.de (134.2.17.166)' can't be esta
ED25519 key fingerprint is SHA256:DpoYXo+4IeCgYwh3+vhZrZdiys3GJjqZ94wVzlADRWQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'login1.tcml.uni-tuebingen.de' (ED25519) to the list of k
stud201@login1.tcml.uni-tuebingen.de's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-138-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon May 19 03:31:56 PM CEST 2025

  System load:  0.55              Processes:               248
  Usage of /:   37.3% of 31.32GB  Users logged in:         5
  Memory usage: 33%               IPv4 address for enp6s19: 134.2.17.166
  Swap usage:   2%


Expanded Security Maintenance for Applications is not enabled.

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

--------------------------------------------------------------------------------
TCML Cluster info
last modified:  2024-10-04
```

Login to the cluster

IMPORTANT: LOGINS NODES ARE **NOT** COMPUTE NODES. YOU CANNOT RUN YOUR PROGRAMS HERE.

Instead, a login node is used to **submit** jobs to the cluster.

The TCML Cluster has 3 login nodes. You can chose which one to use.



Transfer data to the cluster

Dataset:

- You can transfer your data with scp command:

```
scp SOURCE USERNAME@login1.tcml.uni-tuebingen.de:~/
```

NAME top

scp — OpenSSH secure file copy

SYNOPSIS top

scp [**-346ABCOpqRrsTv**] [**-c** *cipher*] [**-D** *sftp_server_path*] [**-F** *ssh_config*] [**-i** *identity_file*] [**-J** *destination*] [**-l** *limit*] [**-o** *ssh_option*] [**-P** *port*] [**-S** *program*] [**-X** *sftp_option*] *source ... target*

DESCRIPTION top

scp copies files between hosts on a network.

scp uses the SFTP protocol over a *ssh*(1) connection for data transfer, and uses the same authentication and provides the same security as a login session.

Code:

- You can copy your code with scp as well, or use GitHub to sync your code (we suggest you the second one).

- Use GitHub's auth token (classic) https://github.com/settings/tokens

Install Conda

- Note: The cluster does not give you access to sudo.
- To install miniconda, download the script from anacoda and run it.
- If, after running the .sh script conda does not work, run . ~/miniconda3/bin/conda init and then re-login

Run the commands and follow the instructions:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

Select yes when running `conda init`, which would append the following lines in `~/.bashrc`:

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/mnt/<storage_disk>/home/<username>/bin/miniconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/mnt/<storage_disk>/home/<username>/bin/miniconda3/etc/profile.d/conda.sh" ]; then
        . "/mnt/<storage_disk>/home/<username>/bin/miniconda3/etc/profile.d/conda.sh"
    else
        export PATH="/mnt/<storage_disk>/home/<username>/bin/miniconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

Relaunch shell to make it work:

```
bash
```

In terminal, it would add a prefix of `(base)`, to indicate that you are in the base env of conda.

Source: https://gist.github.com/Hansimov/2d5d5985116039a0f2976dec91e8ed14

SLURM

- SLURM = **Job Scheduler** for clusters
- You write a **job file** and submit it
- SLURM manages:
  - Resources (CPUs, memory, GPUs)
  - Queues and priorities
  - Job execution and logging

CLUSTER FILE SYSTEM

- The main directory is the **/home** directory. Here, every user has their own folder where they can keep their files.
- A few useful scripts, datasets and singularity recipies can be found in **/common**:
  - **/datasets**: some of the most well-known machine learning datasets.
  - **/share**: here you can share files with other users
  - **/singularityImages**: helpful singularity images and recipes
  - **/userGuides**: more guides and a tutorial script

SUBMIT A JOB VIA SLURM

Step 1:

Create a file like `project1.sbatch`

It's just a **bash script** with special **instructions** for SLURM.

```
stud201@login1:~/slurm_testing$ touch project1.sbatch
```

SUBMIT A JOB VIA SLURM

Step 2:

Fill the file with the configuration for SLURM

```
  GNU nano 6.2                                  project1.sbatch
#!/bin/bash
#SBATCH --job-name=TCML-TEST           # Just a name for the job
#SBATCH --cpus-per-task=4             # Max 24 per node
#SBATCH --partition=day               # day, week, or month
#SBATCH --mem-per-cpu=3G              # Memory per CPU, max 251GB per node
#SBATCH --gres=gpu:1                  # GPUs to use (max 4 per node)
#SBATCH --time=10:00                  # Max time (HH:MM)
#SBATCH --error=job.%J.err            # Error log
#SBATCH --output=job.%J.out           # Output log
#SBATCH --mail-type=ALL               # Email notifications
#SBATCH --mail-user=your@email.com    # Your email

# Your bash commands go here!
slep 100
echo 'done'
```

SUBMIT A JOB VIA SLURM

Step 2:

Fill the file with the configuration for SLURM

```
#SBATCH --gres=gpu:1080ti:4
# to use 4 (the maximum amount!) 1080ti GPUs

#SBATCH --gres=gpu:2080ti:8
# to use 8 (the maximum amount!) 2080ti GPU

#SBATCH --gres=gpu:A4000:1
# to use 1 (the maximum amount is 4) A4000 GPUs

#SBATCH --gres=gpu:L40S:3
# to use 3 (the maximum amount is 8) L40S GPU
```

SUBMIT A JOB VIA SLURM

Step 3:

Submit the job and wait for it to run

```
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
stud201@login1:~/slurm_testing$ sbatch project1.sbatch
Submitted batch job 1507200
```

SUBMIT A JOB VIA SLURM

Step 3:

Submit the job and wait for it to run

```
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
stud201@login1:~/slurm_testing$ sbatch project1.sbatch
Submitted batch job 1507200
```

```
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
stud201@login1:~/slurm_testing$
```

Wait a moment… where is my job?

SUBMIT A JOB VIA SLURM

Step 3:

Submit the job and wait for it to run

```
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST
stud201@login1:~/slurm_testing$ sbatch project1.sb
Submitted batch job 150720
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST
stud201@login1:~/slurm_testing$
```

Code is broken

SUBMIT A JOB VIA SLURM

Step 4:

Your code is broken 100%. Check errors and fix

```
stud201@login1:~/slurm_testing$ ls
job.1507200.err  job.1507200.out  project1.sbatch
stud201@login1:~/slurm_testing$ cat job.1507200.err
/var/spool/slurmd/job1507200/slurm_script: line 14: slep: command not found
stud201@login1:~/slurm_testing$
```

SUBMIT A JOB VIA SLURM

Step 5:

Submit again and start praying

```
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
stud201@login1:~/slurm_testing$ sbatch project1.sbatch
Submitted batch job 1507201
stud201@login1:~/slurm_testing$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1507201       day TCML-TES  stud201  R       0:02      1 tcml-node13
stud201@login1:~/slurm_testing$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1501300   L40Sday personal personal  R   22:37:51      1 tcml-node3
           1507201       day TCML-TES  stud201  R       0:08      1 tcml-node13
           1504491       day Scenario    fauth  R    5:28:15      1 tcml-node37
           1501472     month vep_embe  naegele  R    2:52:05      1 tcml-node18
           1505301      week pd+_trai   raible  R    3:52:51      1 tcml-node21
           1507192      week 235_base   nguyen  R      59:33      1 tcml-node12
           1507191      week 235_base   nguyen  R      59:36      1 tcml-node11
           1507190      week 235_base   nguyen  R      59:39      1 tcml-node10
           1506021      week ML_all_m  naegele  R    2:47:43      1 tcml-node17
           1506020      week ML_all_m  naegele  R    2:47:48      1 tcml-node16
           1506019      week ML_all_m  naegele  R    2:47:54      1 tcml-node20
stud201@login1:~/slurm_testing$
```

SUBMIT A JOB VIA SLURM

Step 6:

Check output

```
stud201@login1:~/slurm_testing$ ls
job.1507200.err  job.1507200.out  job.1507201.err  job.1507201.out  project1.sbatch
stud201@login1:~/slurm_testing$ cat job.1507201.out
done
stud201@login1:~/slurm_testing$
```

```
stud201@login1:~/slurm_testing$ tail -f job.1507201.out
done
```

INTERACTIVE JOBS

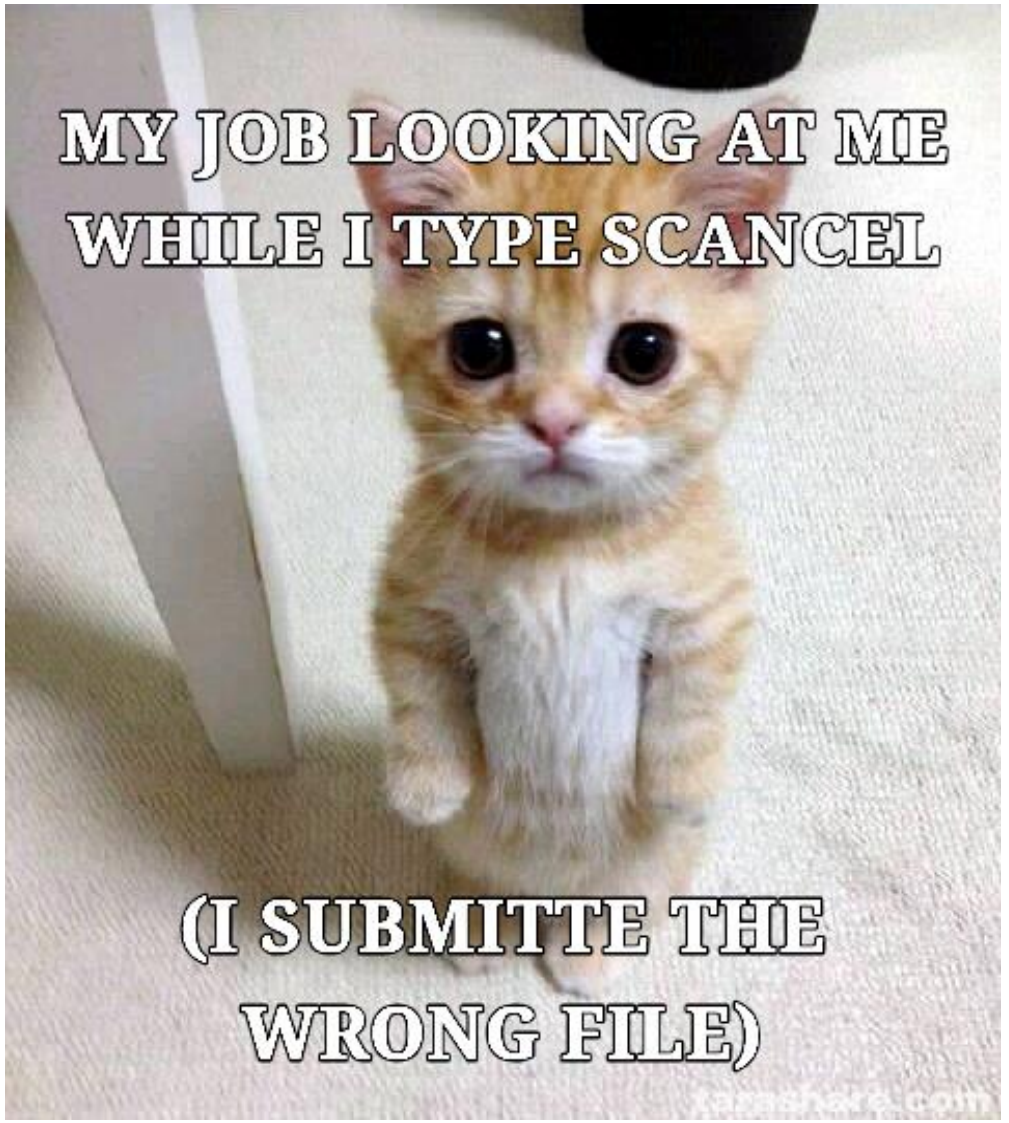
Interactive Jobs are jobs where, instead of launching a program, you ask to create a bash session to use on a cluster node. (use "exit" to end the session)

```
stud201@login1:~/slurm_testing$ srun --job-name "InteractiveJob" --partition=day --ntasks=1 --nodes=1 --gres=gpu:1 --time 1:00:00 --pty bash
stud201@tcml-node16:~/slurm_testing$ nvidia-smi
Mon May 19 17:07:03 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 570.133.20             Driver Version: 570.133.20     CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce GTX 1080 Ti     Off |   00000000:02:00.0 Off |                  N/A |
| 23%   22C    P8              8W /  250W |       3MiB /  11264MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                      GPU Memory |
|        ID   ID                                                             Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
stud201@tcml-node16:~/slurm_testing$
```

CANCEL A JOB

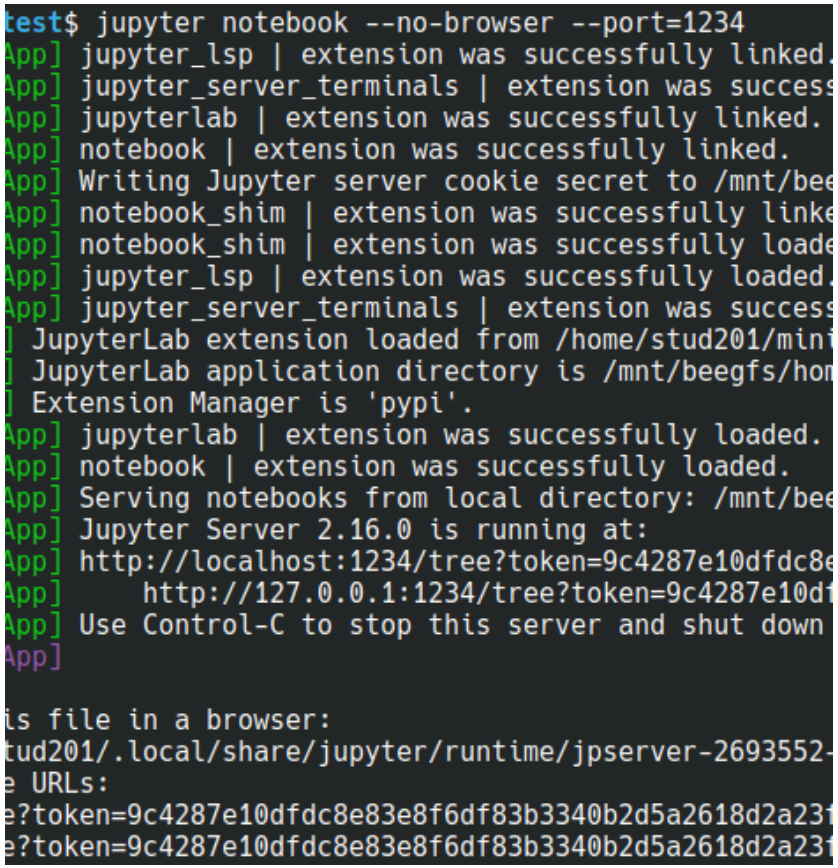
To cancel a job use :

```
stud201@login1:~/slurm_testing$ scancel jobid
```



Run your notebooks from remote

- Once you installed conda, activate an environment and run:
  - pip install jupiterlab
  - pip install notebook
- Then login to a computation node, **NOT ON A LOGIN NODE** (see how to run code on a computation node later), and run:
  - jupyter notebook --no-browser --port=1234
  - Note: you can change the port to the one you prefer
- To connect, open a new terminal on **your** computer and run: ssh -NfL localhost:1234:localhost:1234 youruser@login1.tcml.uni-tuebingen.de
- Finally, open your browser and open http://localhost:1234
- From ther, you can use Jupyter to run your notebooks



PARTITIONS

5.4.2 Partitions

Partition Name	Time limit	Number of nodes
test	15 minutes	3
day	24 hours	33
week	7 days	22
month	30 days	10
L40Sday	24 hours	4

Notice: Any job that exceeds the time limit of its partition will get canceled. Check the details for which node belongs to which queue with `sinfo`.

Final Tips

Before contacting admins, check the FAQ!

- Many problems are already answered
- Saves time for everyone

Best practice:

- Start with small jobs
- Monitor memory/time usage
- Gradually scale up

Misc:

- If the gpus are not used 100% all the time, you have a bottleneck somewhere
- Use iteractive sessions to test your code before sending a task