bw|HPC

2nd bwUniCluster & DACHS User Workshop

HPC @ HAW
29.07.2025

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

UNI FREIBURG

UNIVERSITÄT HOHENHEIM 1818

HOCHSCHULE ESSLINGEN

Universität Konstanz

UNIVERSITÄT MANNHEIM

Universität Stuttgart

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

KIT Karlsruher Institut für Technologie

ulm university universität ulm

Förderträger: Baden-Württemberg MINISTERIUM FÜR WISSENSCHAFT, FORSCHUNG UND KUNST

Overview

The **slides** will be available as PDF, a **recording** available for download on:

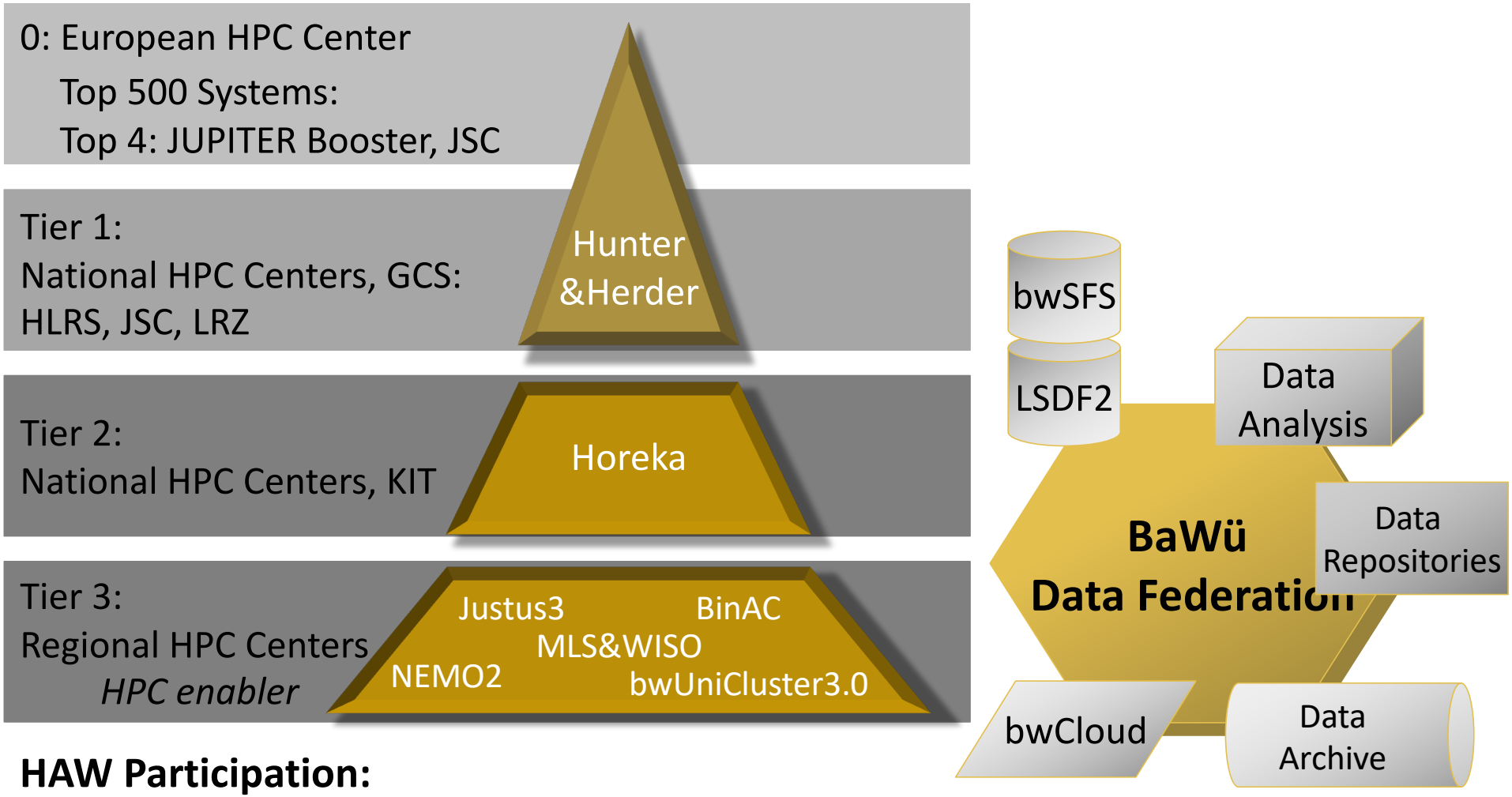
https://www.hs-esslingen.de/informatik-und-informationstechnik/forschung-labore/forschung/laufende-projekte/dachs. (with info on how to register, log in, etc.)

Today‘s topics:

- Repetition of Project Overview
- Usage of Software Modules
- Best Practices Microsoft Visual Studio Code (VScode)
- Forschungsdatenmanagement (FDM) with DACHS
- Security-Info: Firewall on DACHS Login
- Parallel Filesystems Lustre/BeeGFS & localscratch
- How to acknowledge bwUniCluster / DACHS in publications
- Further documentation

HPC Overview

Strategy for implementing High Performance Computing (HPC), Data intensive Computing (DIC)



HAW Participation:

1. Partnering HPC@HAW: Share of bwUniCluster (HAW as-a-whole)
2. Partnering Datenanalyse Cluster der HS (DACHS)

Project HAW Datenanalyse Cluster BaWü

- Partnering as an Assocation with a cross-site installation:

1. HS Aalen
2. HS Albstadt-Sigmaringen
3. HS Esslingen
4. HS Heilbronn
5. HS Karlsruhe
6. HTWG Konstanz
7. HS Mannheim
8. HS Offenburg
9. HS Reutlingen
10. HfT Stuttgart
11. THU Ulm

Application as “Großgeräte der Länder”, reviewed positively by DFG and 50% co-funded by MWK and all partners.

Setup Datenanalyse Cluster BaWü

The Hardware

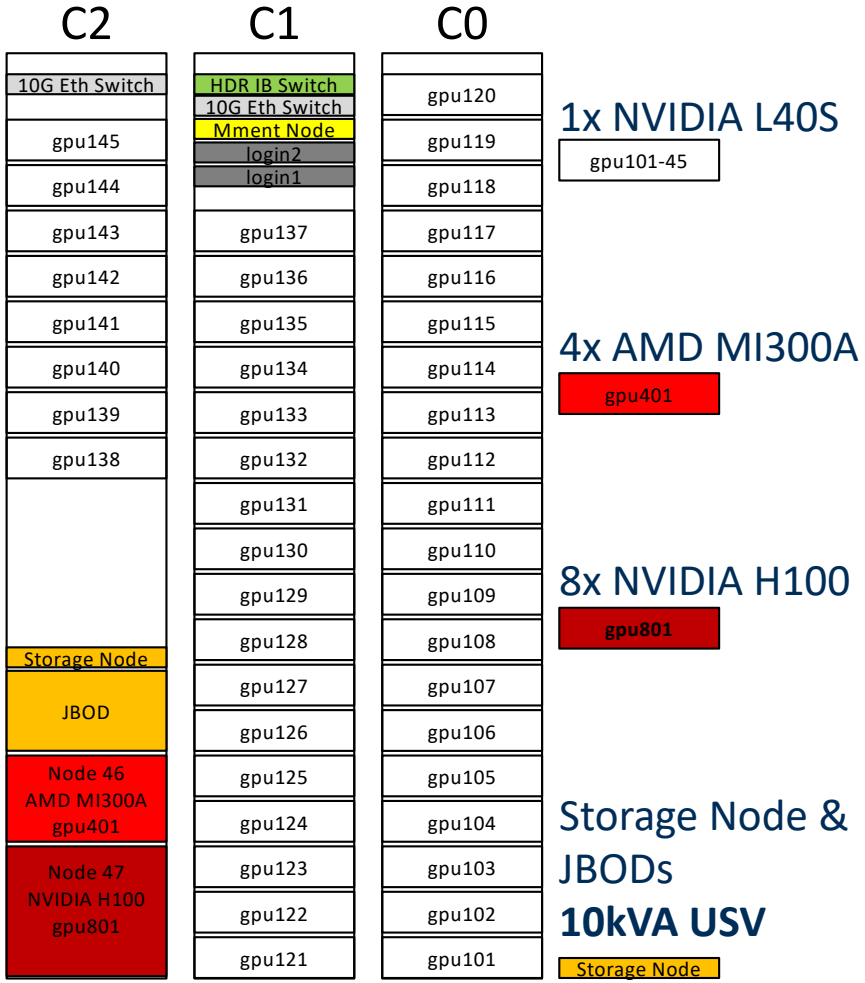
- **45** x single GPU nodes (NVIDIA L40S á 48GB)
- **1** x Quad-socket APU node
  i.e. 4x AMD MI 300A, total 512 GB HBM3 RAM
- **1** x Octo-GPU node (8x H100 á 80GB SXM5),
  Dual-AMD EPYC 9**4**54, i.e. 48 cores, 128 MB L3
  total 1,5 TB ECC-RAM
- **2 x Login** and 1x Management node
- Parallel BeeGFS filesystem with **700 TB (netto)**
- NVIDIA/Mellanox Switch IB HDR 200GBit

All nodes with:

- Dual-AMD EPYC 9254 CPU
  i.e. 24 Cores, 2.9 GHz, 128 MB L3
- 384 GB ECC RAM
- 1,92 TB local SSD (for local scratch!)

Using the available cooling infrastructure & racks.

In total **75kW peak** cooling requirement



Usage of Software Modules

Software Modules 1/3

- On HPC commonly used software is provided centrally upon request
- On bwUniCluster & DACHS we use Lmod to make Software available:
  - `module avail`
    shows all available software, e.g. `module avail compiler` shows just these
  - `module load compiler/gnu/15.1`
    loads this specific compiler version, while `module load compiler/gnu/` loads the latest available gnu compiler (here it **is** v15.1)
  - `module help mpi/openmpi` and `module whatis module_name`
    shows information on the module, e.g. where to look for examples, etc.
  - `module purge`
    resets the environment, unloads any modules
- SW Environments *just* adapt your `PATH`, `LD_LIBRARY_PATH`, `MANPATH` and set other environment variables, e.g. the compiler above sets `GNU_VERSION` and `GNU_HOME`.
- Check these variables prior and after loading with `env|sort|less`

Software Modules 2/3

- To make sure You use the proper Software, check the env-var PATH:
  `echo $PATH`
  e.g. that the `gcc` compiler is from `compiler/gnu/15.1`:
  `which gcc`
- SW modules may be „**hidden**“, aka not shown in `module avail`
  This module‘s name is preceded with a ‘.‘ and can still be loaded:
  `module load devel/pocl/.6.0` (Portable OpenCL v7 is avail)
- If the SW version is not important, e.g. You want the latest, skip it:
  `module load compiler/gnu`
- Software has dependencies (on system libraries, other modules).
  These are loaded automatically, e.g.
  `module load devel/python/3.13.3-llvm-18.1`
  of course, this can lead to annoyingly long names

Software Modules 3/3

- The bwUniCluster and DACHS Cluster run up-to-date Rocky Linux
- Most SW for every-day usage is available and recent, e.g. `cmake`, `git`
- If You need further commonly used SW, please let us know!

- Any site may install SW into their organizations SW directory:
  `/opt/bwhpc/es/…`
  containing the **same** SW categories available in `common/`:

`admin`	`mpi`
`cae`	`numlib`
`compiler`	`system`
`cs`	`vis`
`devel`	and a corresponding `modulefiles`

- On bwUniCluster3.0, there additionally is EasyBuild to provide SW.

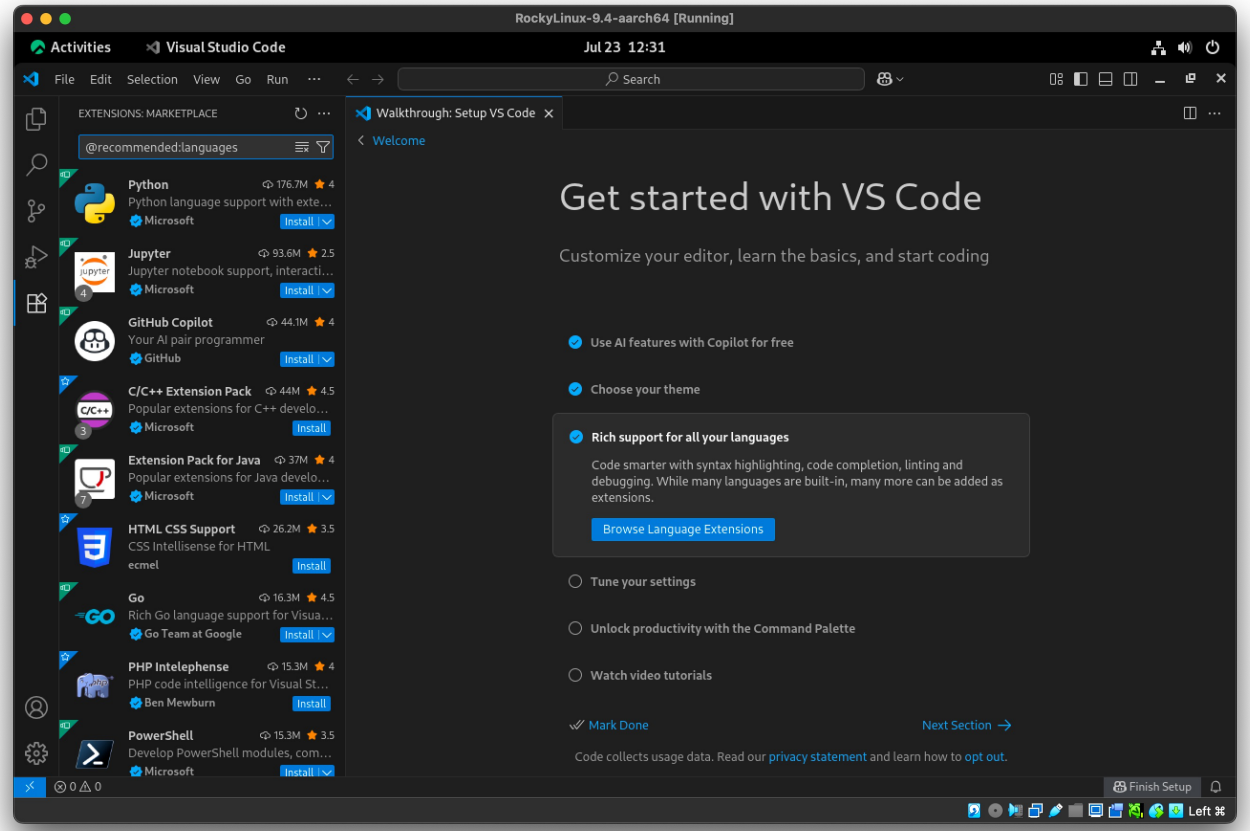
Build Software?

- How to configure:
  - cmake
  - AutoConf?
  - Scons, Bazel
- How to build:
  - Make
  - Ninja
- Etc. is very SW-dependent – if interested → bwHPC TigerTeam
  Please talk to us.

Visual Studio Code

Visual Studio Code 1/3

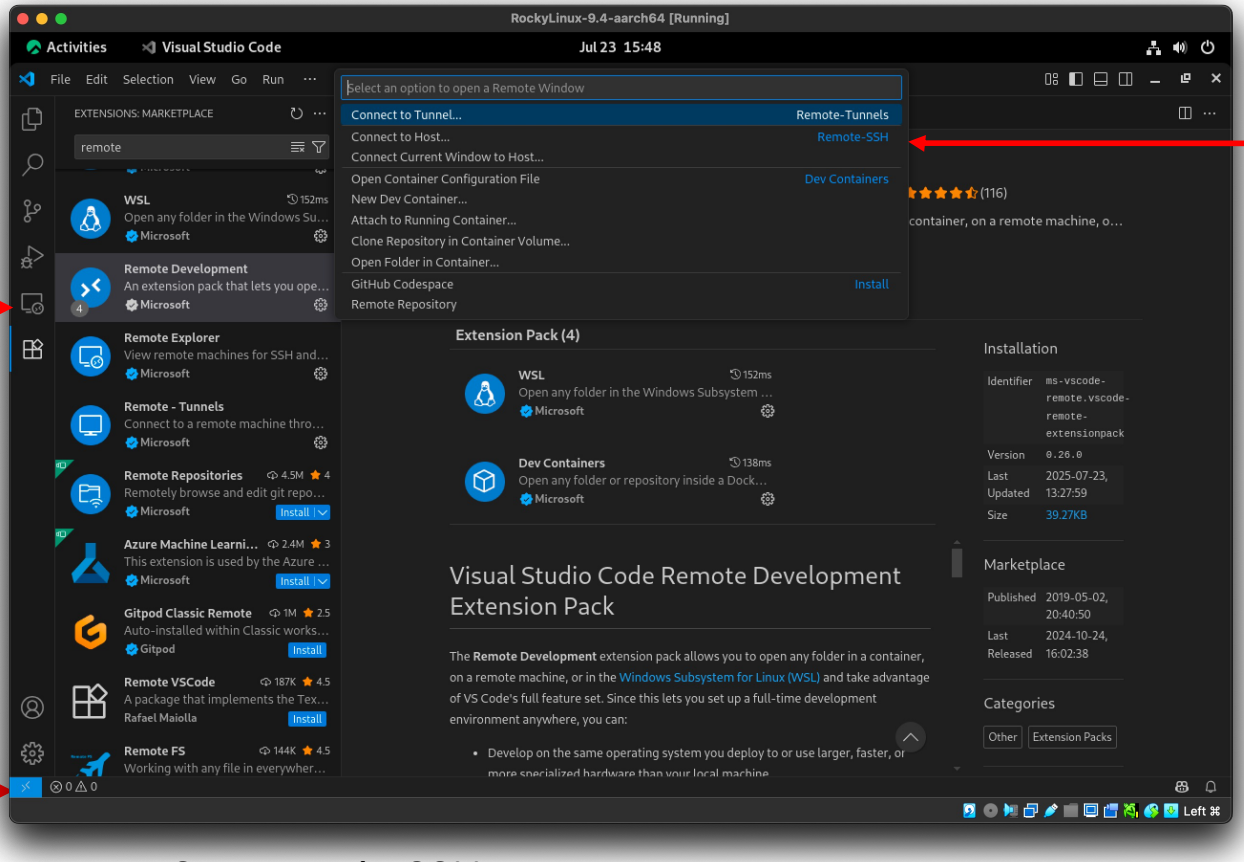
- MS Visual Studio Code is a great Open Source tool with Extensions:



- Integrated Development Environment
- Code completion for many languages (C, C++)
- GitHub CoPilot for AI pair programming

Visual Studio Code 2/3

- Extensions “C/C++” with IntelliSense and “Remote Development”:



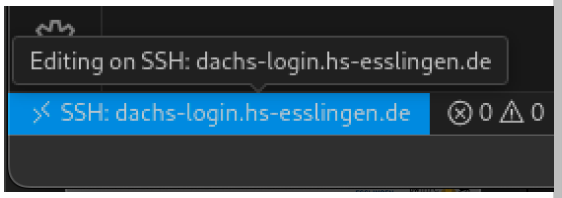
New tab: “remote explorer”

- Access remote Servers via SSH
- Select “Connect to Host” and configure one of Your SSH connections

Visual Studio Code 2/3

- Adding an SSH host adapts Your `~/.ssh/config`:

```
Host dachs
      HostName dachs-login.hs-esslingen.de
      User es_rakeller
```



- After entering OTP and Password You're logged in:
  (in your remote `~/.vscode-server/` ca. 450MB are installed)
- You have a Terminal and may edit Files open Folders (Directories)

- SSH offers so much more.

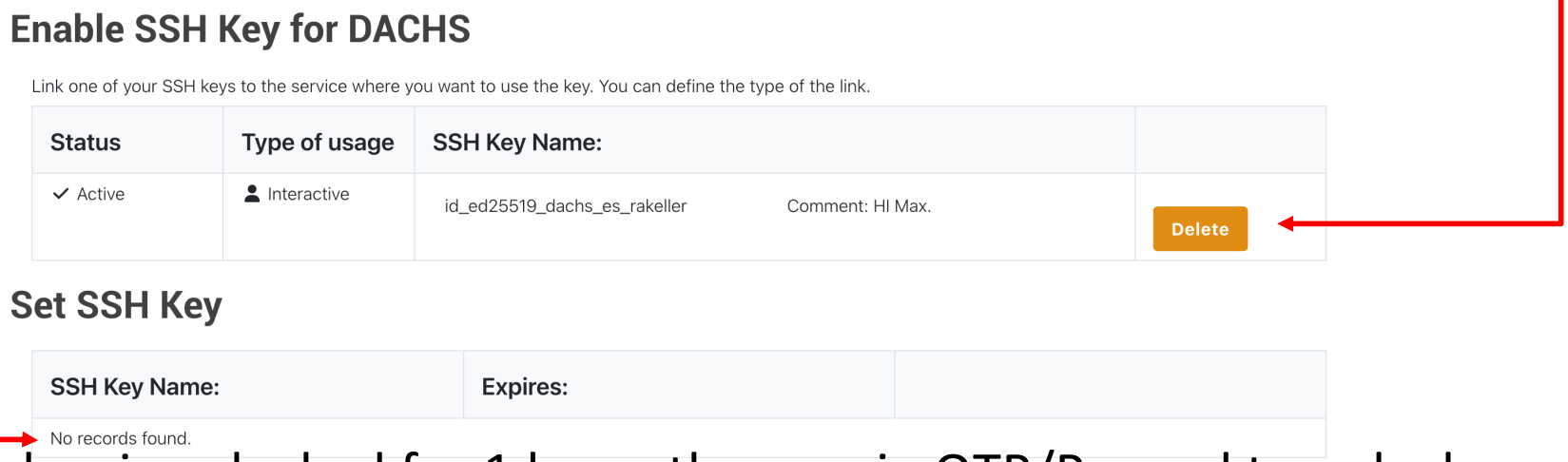
ssh Jump Host

- Instead of logging directly into any server, You may need a jump host in case of not using VPN/BelWue Firewall restrictions (change *italic*)
- In `~/.ssh/config`:

```
Host bwJUMP
    User ORG_USERNAME
    HostName bwunicluster.scc.kit.edu
    ProxyCommand /usr/bin/ssh -i
~/.ssh/id_rsa_bwcloud_proxyjump -l ubuntu -v -W '[%h]:%p'
YOUR.bw-cloud-instance.org
    IdentityFile ~/.ssh/id_rsa_bwcloud_proxyjump
    ForwardAgent yes
    PubkeyAuthentication yes
Host bwcloud
    User ubuntu_username_on_bwcloud
    HostName YOUR.bw-cloud-instance.org
    IdentityFile ~/.ssh/id_rsa_bwcloud_proxyjump
    ProxyJump bwJUMP
```

ssh Public Key Access (see last time)

- Instead of typing OTP & passwd all the time, use SSH key with passwd
  `ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519_dachs_`*`USER`*
- Add this into `~/.ssh/config` for your service as:
  `IdentityFile ~/.ssh/id_ed25519_dachs_`*`USER`*
- Register Your **public Key** in https://login.bwidm.de in Tab Index → My SSH PubKeys → Add Key – it then shows below:
- In Tab „Registered Service“ → “Add Key” to Service ; afterwards:



- This key is unlocked for 1 hour, then again OTP/Passwd to unlock
- Use `ssh-agent` (to store Keys) & `ssh-add` to cache above Key!

VS Code remote in combination with git

- In combination with a dual-screen solution:
  Using VS-Code locally on 1st Screen – check into git
  Using VS-Code remote on 2nd Screen – to check out and compile

  With possible a 3rd Screen for Documentation/Browser, etc.
- Offers a good turn-around time for trying out new features...

- Good Input from R. Breuer: There is VScodium without MS branding

Forschungsdatenmanagement (FDM) with DACHS

Publication of Your research data

- There are different services to:
  - Temporarily Store large-scale data at cost (SDS@HD)
  - Backup cold data on long-term storage (bwDataArchive)
  - Repositories for specific Science Communities may use bwSFS and bwSFS2 (soon)

- In need to publicize Your research paper's data? Use DACHS' FDM:

```
fdm-publish [-h] -b BIBTEX -a ARCHIVES1 [ARCHIVES2 …]
```

Optional Parameters []
Here to provide help

Must provide a
Bibtex File

And one (or multiple
archives to download

- It generates a HTML and provides this/these archive(s) in a publicly accessible location, e.g. the ROVER dataset by Fabian Schmidt et al: https://fdm.hs-esslingen.de/schmidt2025rover/

FDM Publication Internals

- Agreement:
  - Storage: ~1TB/Prof.
  - Accessible via this URL: ~5 yrs
  - No QoS: no backup, best-effort
- The URL‘s name is provided by key your `.bib`-file:

```
es_makunzel@login2 ~/fdm.prod $  fdm-publish -b mypaper.bib -a data.tgz
Copying: 100%|#####################################################| 10.0G/10.0G [02:39<00:00, 62.7MB/s]
Data successfully published.
You can adjust and edit your index.html located at  /beegfs/fdm/Mustermann2025/index.html
Please write this path down, so you can edit the file in the future.
You can access your data at  https://fdm.hs-esslingen.de/Mustermann2025/
es_makunzel@login2 ~/fdm.prod $  cd /beegfs/fdm/Mustermann2025/
es_makunzel@login2 /beegfs/fdm/Mustermann2025 $  ls -lh
total 9.0G
-rw-r--r-- 1 es_makunzel nginx 9.4G Jul 26 10:30 Mustermann2025.tgz
-rw-r--r-- 1 es_makunzel nginx  914 Jul 26 10:27 index.html
```

- The archives may be combinations of `.tar*`, `.gz` and `.zip`
- The HTML and archives are copied into `/beegfs/fdm/YOUR_KEY`, it is **writable by You**, You may amend archives HTML **later live** at will

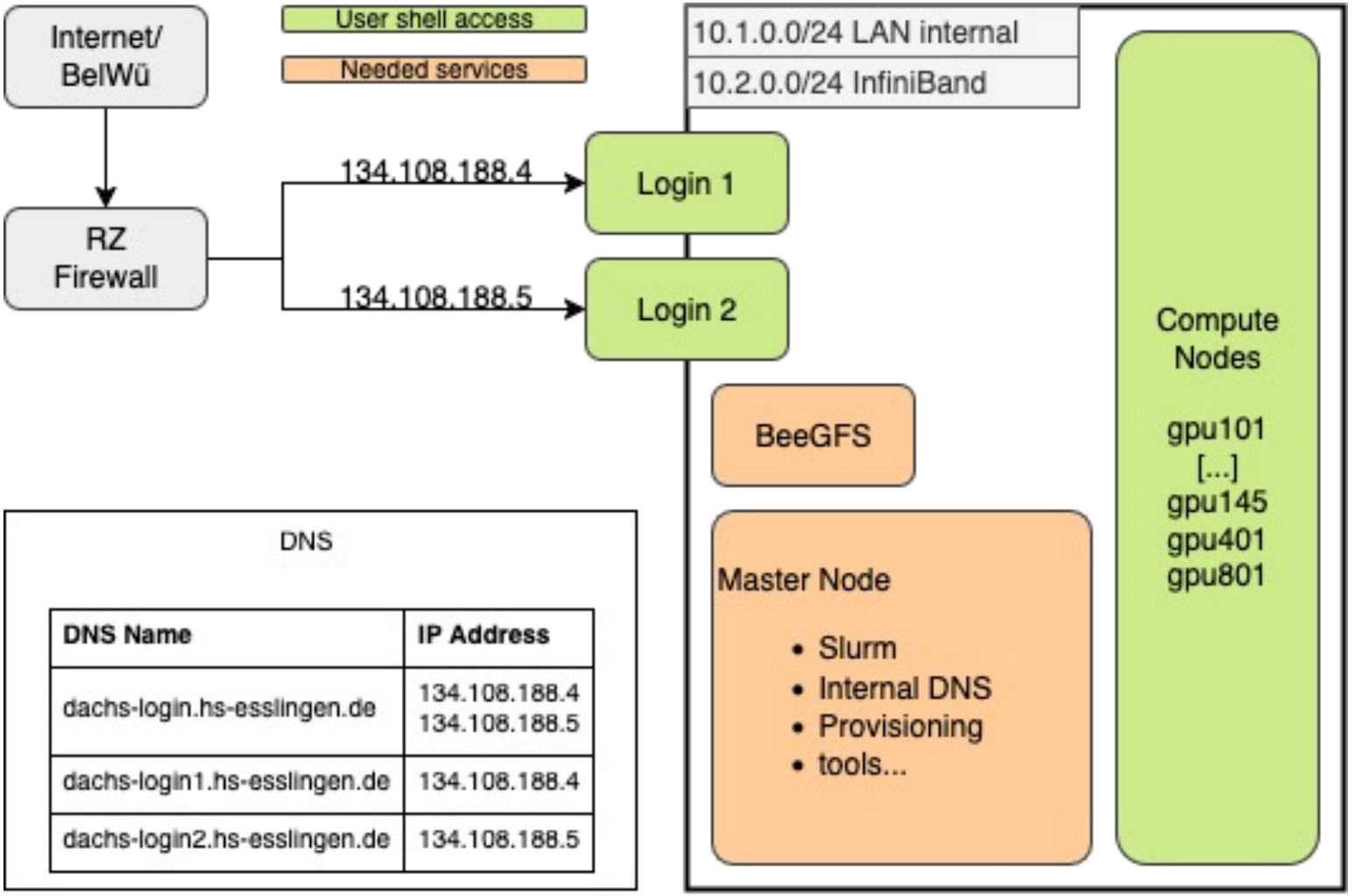
Security-Info:
Firewall on DACHS Login

Firewall Rules – Overview

- Changed default policy for incoming traffic from ACCEPT to DROP with exceptions:
  - Allow SSH (external and internal to `134.108.188.{4, 5}`)
    (Our firewall limits external IPs to BelWue IP ranges & internal: hs-esslingen.de)
  - Allow HTTPS (for FDM worldwide – static HTML serving)
  - Allow UCARP (Common Address Redundancy Protocol) between Login Nodes: dachs-login.hs-esslingen.de provides two IPv4 addresses
  - Allow Slurm communication (`slurmctld`, `slurmd`, compute ↔ login nodes)
  - Allow BeeGFS communication (metadata-server from cluster internal IPs)
  - ICMP
- Forwarding traffic (compute nodes ↔ internet)
- NAT (Network Address Translation) for the compute node subnet

Firewall Rules – Cluster Overview

- Graphically:



DNS Name	IP Address
dachs-login.hs-esslingen.de	134.108.188.4 134.108.188.5
dachs-login1.hs-esslingen.de	134.108.188.4
dachs-login2.hs-esslingen.de	134.108.188.5

Firewall Rules – Log dropped traffic

- Show dropped packets. Add rule to the end of the `INPUT` chain

```
-A INPUT -j LOG --log-prefix "iptables:policy drop: " --
log-level 6
```

- Load the new ruleset
- Grep by the log prefix:

```
grep 'iptables:policy drop' /var/log/messages
```

Example:

-
```
Jul 26 09:49:30 login1 kernel: iptables:policy drop:
IN=bond-ext OUT=
MAC=44:49:88:02:f3:98:e4:1f:7b:eb:7b:9f:08:00
SRC=45.227.254.156 DST=134.108.188.2 LEN=52 TOS=0x00
PREC=0x00 TTL=117 ID=3751 DF PROTO=TCP SPT=65179 DPT=443
WINDOW=200 RES=0x00 CWR ECE SYN URGP=0
```

Parallel Filesystem & local scratch

Workspace Tools Repetition

- Home & Workspaces are on Lustre (bwUniCluster) / BeeGFS (DACHS)
- Workspaces allow large, fast but temporary storage
- Default duration 30 days (extendable 3 times, up to 90 days each)
- Basic commands
  - Create: `ws_allocate <name> <days>`
  - Extend: `ws_extend <name> <days>`
  - Delete: `ws_release <name>`
  - Find storage path: `ws_find <name>`

```
$ ws_find test_workspace          # shows the directory
/beegfs/scratch/workspace/xx_use-test_workspace
```

  - List your workspaces: `ws_list`

Workspace Tools

- Email reminder start being send **1 week** before expiry
- After expiry workspace is kept for another 14 days, only **then** deleted
  - Restore using `ws_restore`

```
$ ws_restore -l                    # list restorable workspaces

$ ws_allocate <new-ws>             # allocate a new workspace

$ ws_restore <old> <new-ws>        # restore the under new name
```

- To share a workspace with other users on the system (based on ACLs):

```
$ ws_share <ws> <user>             # make available to other users
```

More examples in the user guide
https://github.com/holgerBerger/hpc-workspace/blob/master/user-guide.md

Local Scratch

- Every node has ~1TB on a NVME SSD for user jobs!
- XFS file system for Users mounted under `$TMPDIR`, which points to: `/localscratch/tmpdir.${SLURM_JOB_ID}`
- Use if your programs needs to write and/or read frequently from disk!
- Especially many small files will be better put onto these

Examples:

- Copy (recursively) a directory from your `$HOME` there:
  ```
  cp -r $HOME/dir ${TMPDIR}
  ```
- Unpack file from Workspace into directory:
  ```
  unzip `ws_find my_workspace`/file.zip -d ${TMPDIR}
  ```

- Files in `${TMPDIR}` are **deleted** after your job terminates!
  In Your batch script, please **copy** back results into your `$HOME`!

Acknowledgements & further Documentation

Acknowledgements of Usage of HPC Systems

- Future funding will depend on the systems being used.
- We are dependent on being visible and are held accountable.
- Therefore we ask you, to please acknowledge the Clusters, e.g. by referencing them in project proposals, and state the usage as in:

bwUniCluster:

“The authors acknowledge support by the state of Baden-Württemberg through bwHPC.”

DACHS:

“We thank the DACHS data analysis cluster, hosted at Hochschule Esslingen and co-funded by the MWK within the DFG's "Großgeräte der Länder" program, for providing the computational resources necessary for this research.”

Further Documentation

- bwHPC Wiki:
  https://wiki.bwhpc.de
  https://wiki.bwhpc.de/e/bwUniCluster3.0
  https://wiki.bwhpc.de/e/DACHS
  BeeGFS Documentation
  Lustre Documentation
- Training Platform:
  https://training.bwhpc.de
- In case of problems create a Ticket:
  https://www.bwhpc.de/supportportal/
- Or send an email to dachs-support@hs-esslingen.de or dachs-admin@hs-esslingen.de

- Next Workshop: 7th of November 2025 at 2pm

Fragen?

Bei weiteren Fragen: dachs-admin@hs-esslingen.de

Oder ein Ticket auf: https://www.bwhpc.de/supportportal