



Introduction to Atos HPCF and ECS

Roberto Cuccu

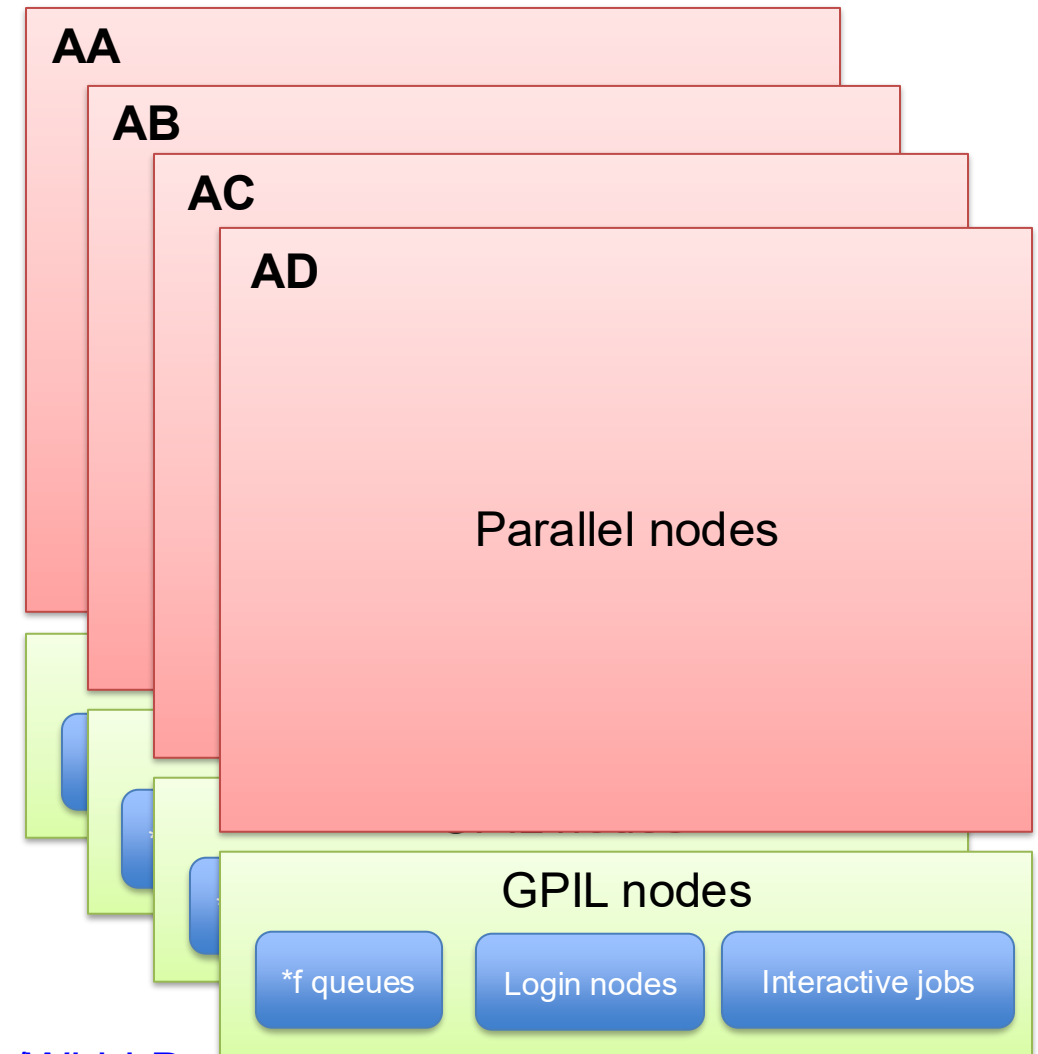
roberto.cuccu@ecmwf.int



ECMWF AtoS HPCF



- 4 Complexes
 - Two in each Computer Hall
 - Each Complex consists of two partitions:
 - Parallel:
 - 1920 nodes for parallel compute
 - exclusive access (only one job in a node)
 - General Purpose (GPIL)
 - 112 nodes for general purpose use
 - » More memory, local SSD
 - Shared access (many jobs in the same node)
- One Slurm scheduler in each complex
- Same Filesystems and applications everywhere

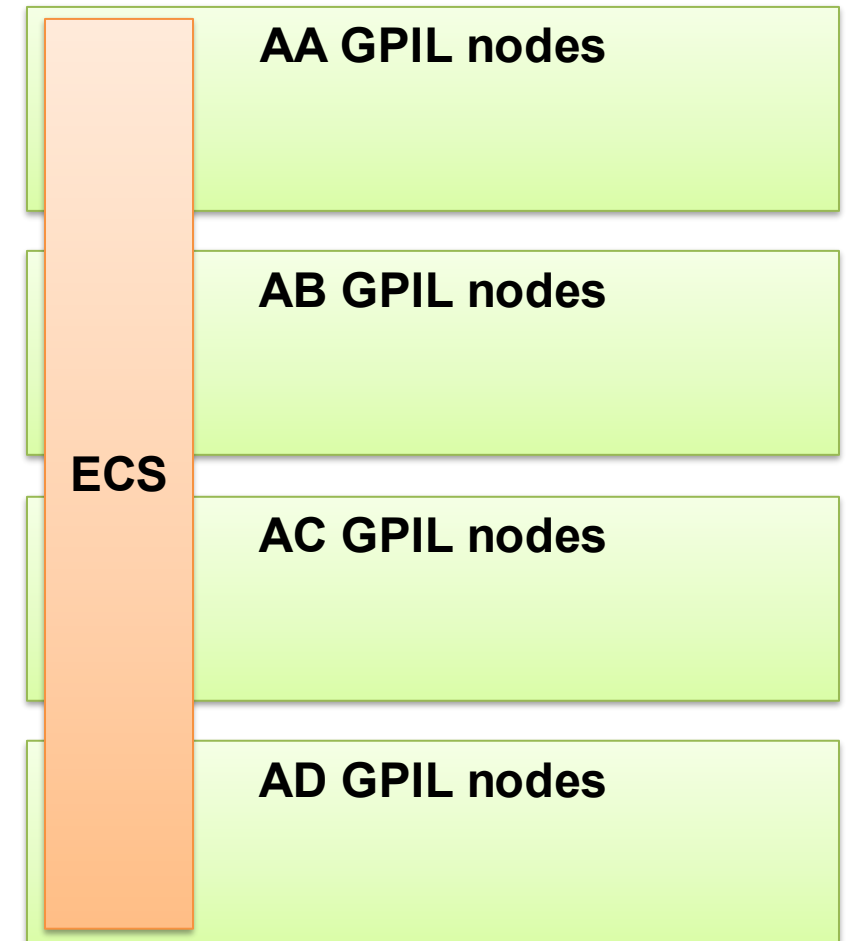


<https://confluence.ecmwf.int/x/WhhbDg>



ECS – ECGATE Class Service

- For users with no formal HPCF access
 - Including those from Cooperating States
- “Virtual” complex with nodes from 4 complexes
- Same Apps and Filesystems as main complexes
- **Independent Slurm Batch system**
- **Serial or very small parallel workloads**
- **No SBU billing**



<https://confluence.ecmwf.int/x/WhhbDg>

Where to start



 **ECMWF** Spaces ▾

Search

Log in

PAGE TREE

Support portal

Login and authentication informatic

Teleport SSH Access

Atos HPCF and ECS services

HPC2020 User Guide

HPC2020: How to connect

Atos HPCF: System overview

HPC2020: Shells

HPC2020: Filesystems

HPC2020: File transfers

HPC2020: Software stack

HPC2020: Batch system

HPC2020: Cron service

HPC2020: Using ecFlow

HPC2020: Accounting

HPC2020: ECaccess

HPC2020: Time Critical option

HPC2020: Time Critical Option

HPC2020: GPU usage for AI an

HPC2020: Missing features an

HPC2020: FAQs

Atos HPCF training material

Space tools

Pages / User Documentation / Atos HPCF and ECS services

HPC2020 User Guide

Created by Xavier Abellan, last modified on Nov 01, 2023

This is the User guide for the [Atos Sequana XH2000 HPCF](#), installed in ECMWF's data centre in Bologna. This platform provides both the HPCF (**AA**, **AB**, **AC**, **AD** complexes) and ECGATE services (**ECS**), which in the past had been on separate platforms.

✓

Introductory tutorial
If you are new to the Atos HPCF and ECS services, you may also be interested in following the [Atos HPCF and ECS Introduction Tutorial](#) to learn the basic aspects from a practical perspective.

Below you will find some basic information on the different parts of the system. **Please click on the headers or links to get all the details for the given topic.**

HPC2020: How to connect - [read more]

From outside ECMWF, you may use [Teleport](#) through our gateway in Bologna. For all the details of this connection method please see the [Teleport](#) documentation, where you will find how to best configure your SSH settings. Once the setup of the Teleport client is complete on your end, you will be able to connect with:

Atos HPCF: System overview - [read more]

The Atos HPCF consists of four virtually identical complexes: AA, AB, AC and AD. In total, this HPCF features 8128 nodes:

7680 compute nodes, for parallel jobs

448 GPIL (General Purpose and Interactive Login) nodes, which are devised to integrate the interactive and post-processing work from older platforms such as the Cray HPCF, ECGATE and Linux Clusters.

HPC2020: Shells - [read more]

News Feed

2025-05-21 Change of default versions of ECMWF and third-party software packages - May 2025

When?

The changes took place on **Wednesday 21 May 2025 10:00 UTC**

Do I need to do anything?

2025-03-05 HPCF and ECS Service unavailable for general use

Due to an update required on the Lustre filesystem serving HPCPERM, SCRATCH as well as fws4, the ECMWF HPCF were unavailable to run any standard jobs on:

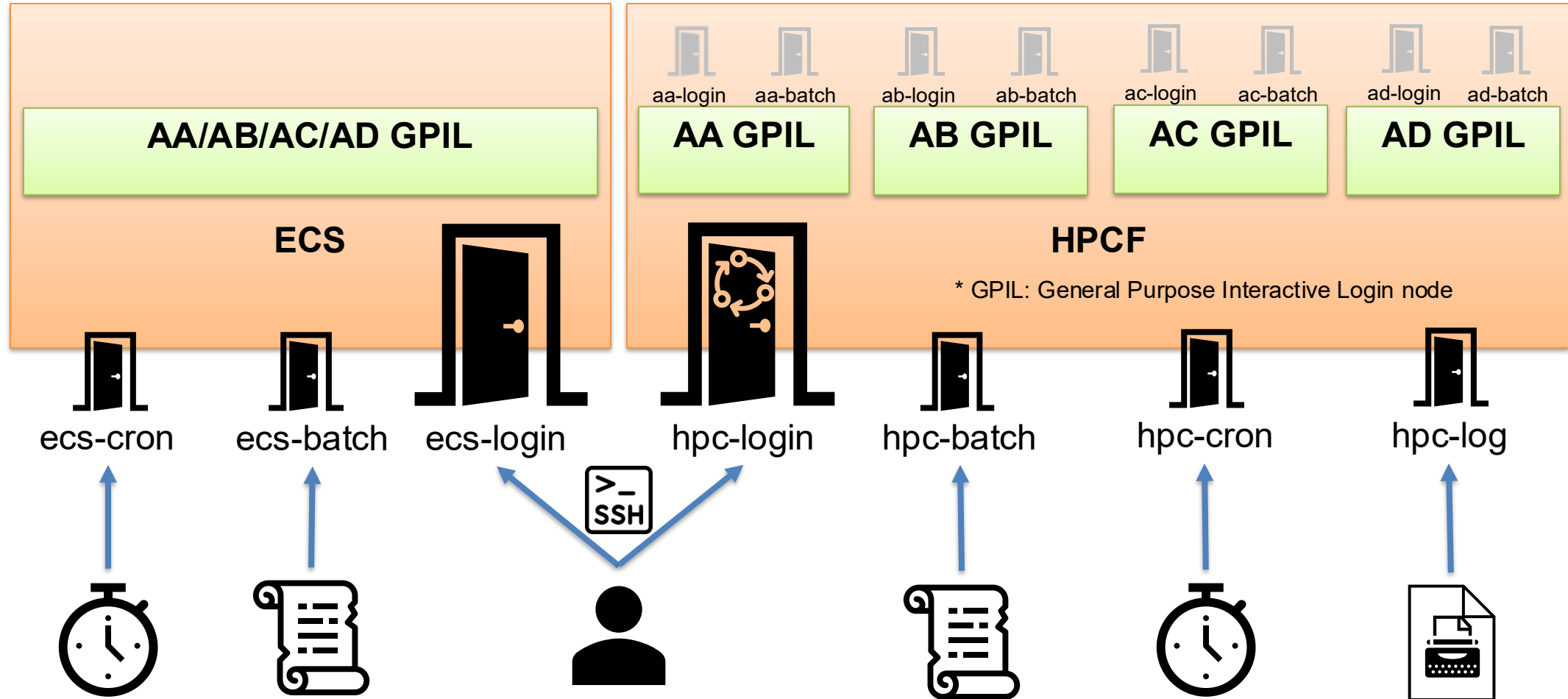
Wednesday 2025-03-05 from 08:30 to 16:30 UTC

During the session, queued jobs were kept on hold with the reason "Licenses", and started once the session is over. At the same time, there was a shorter,

 **ECMWF** EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

4

Which door to use?



<https://confluence.ecmwf.int/x/WBhbDg>



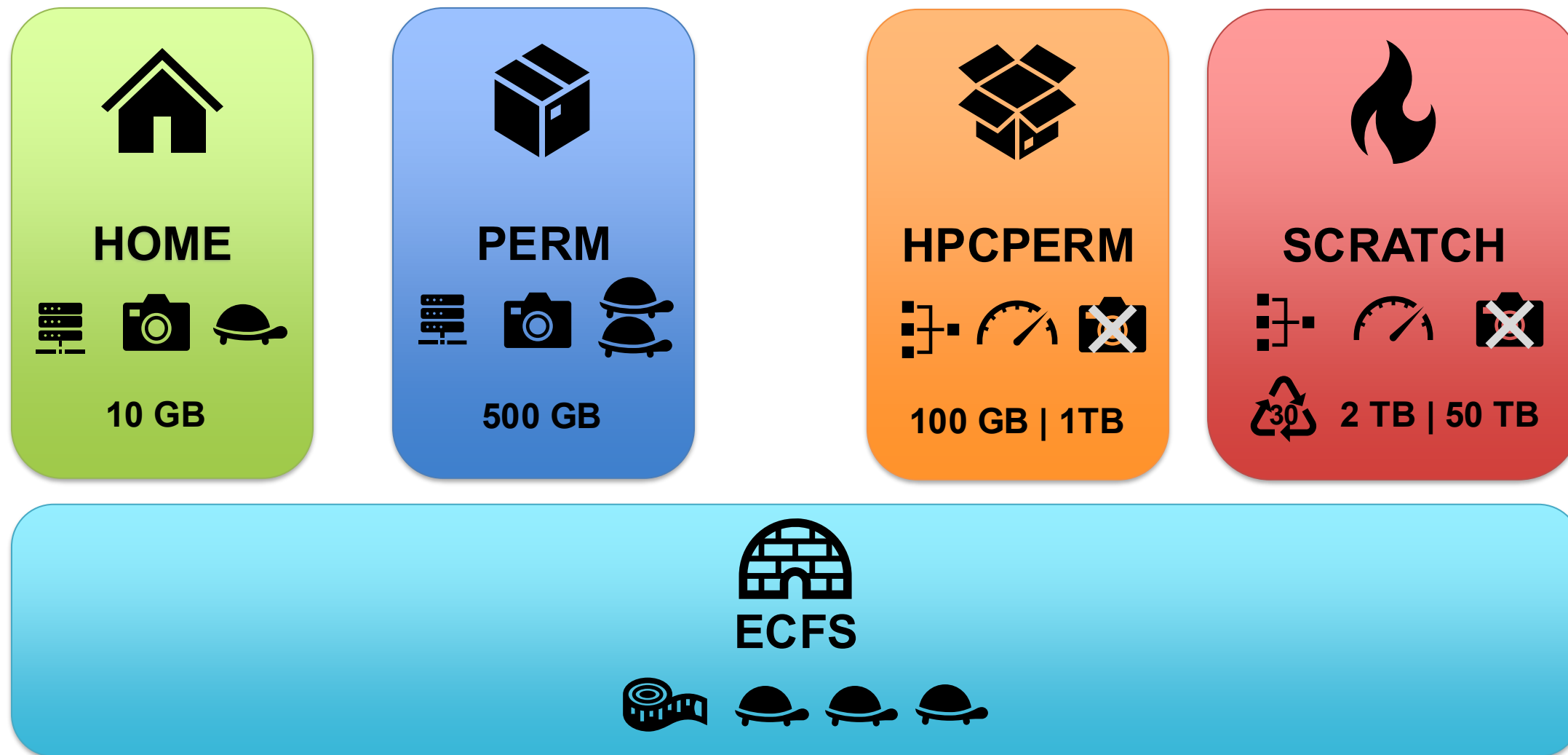
Shells

- Available login shells are **BASH** (default for new users) and **KSH**
- We can change your default shell: let us know via the [ECMWF Support Portal](#)
- You may customise your shell environment with the standard files:
 - ~/.profile and ~/.bash_profile
 - ~/.bashrc and ~/.kshrc
 - ~/.bash_logout



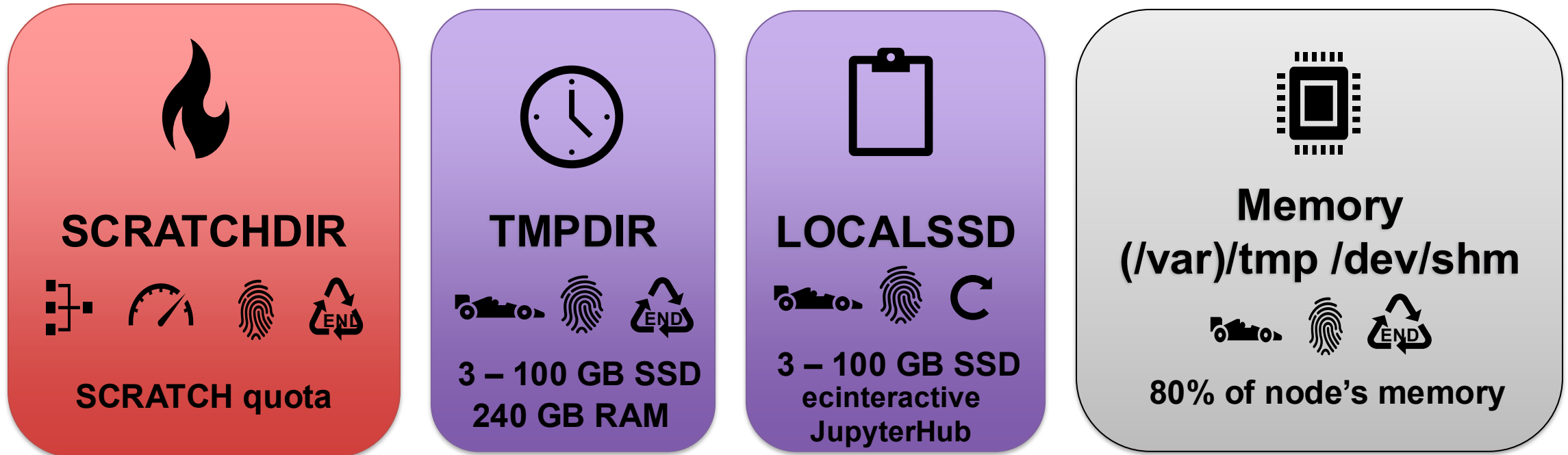
<https://confluence.ecmwf.int/x/WxhbDg>

Storage



<https://confluence.ecmwf.int/x/XBhbDg>

Temporary Storage



Use `quota` command to check your usage & limits

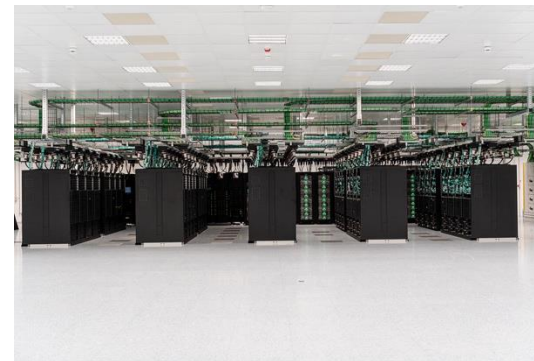
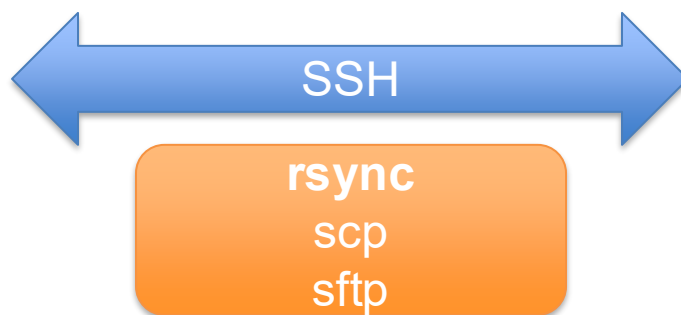
`HOME`, `PERM`, `HPCPERM`, `SCRATCH`, `SCRATCHDIR`, `TMPDIR`, `LOCALSSD` are defined as env variables

<https://confluence.ecmwf.int/x/XBhbDg>



File transfers

- Any tool using OpenSSH behind the scenes should work
 - Teleport should be configured appropriately if running from outside ECMWF



```
user@yourlaptop:~> rsync -avz mydataset hpc-login:/scratch/user/  
user@yourlaptop:~> rsync -avz hpc-login:/scratch/user/mydataset .
```

- SFTP to boaccess.ecmwf.int
- ectrans may be used to push data from HPC to a remote site
- bbcp also possible for advanced use cases.

<https://confluence.ecmwf.int/x/XRhbdG>



Environment Modules

- Manage your software environment

- Over 400 modules available

- Basic commands:

- module load / unload
- module list
- module avail / spider
- module show / help
- module reset

```
$ module overview
----- /usr/local/apps/modulefiles/lmod/prgenvs -----
conda (3)  prgenv (8)  tykky (2)

----- /usr/local/apps/modulefiles/lmod/core -----
R (5)      doxygen (4)  geff (2)      iver (5)      nco (4)      preplm (37)  sqlitebrowser (2)
aec (6)     ecaccess (1)  geocat (4)    jasper (4)    ncview (3)   product-delivery (1)  stat (2)
ai-models (17)  ecbind (2)  geos (5)     java (2)     nexhub (7)   profile_browse (4)  tctracker (4)
aifs-ens-crps (4)  ecbuild (12)  geotiff (2)  julia (3)    ninja (4)    proj (7)      texlive (4)
aifs-single-mse (3)  ecflow (24)  ggo (1)     kdeapps (1)  node (4)     psutils (1)    tflite (3)
aifs-single-nudge (1)  eclaunch (2)  git (6)     lcov (3)     numdiff (1)  pycharm (5)    troika (1)
amduprof (2)  ecml-toolbox (2)  gitflow (1)  lftp (2)     nvtop (1)   pyg2p (18)    udunits (2)
ansible (5)   ecmp (2)        github-cli (2)  libflame (1)  obstat (3)  pypy (2)      uv (1)
antlr (1)    ecps (1)        gmt (4)      libfyaml (1)  octave (5)  python3 (5)   ver0d (14)
apptainer (4)  ecperf (3)  gnuparallel (5)  libspatialindex (2)  ocxenv (8)  qgis (4)      vim (1)
bison (1)    ecprof (3)     go (6)       libtensorflow (3)  onnxruntime (3)  qhull (1)     vtb (13)
blis (3)     ecquote (2)    gperf (1)    lisflood (4)  openjpeg (4)  qt (3)        webdev (7)
bufrdc_tables (2)  eigen (3)     grace (1)    lua (1)       panoply (5)  quaver (15)   wellies (7)
cdftools (2)  esmf (4)       grads (1)    matlab (2)    patchelf (2)  rclone (4)    wgrib (3)
cdo (8)      fcm (2)        grass (5)    maven (4)     pcraster (3)  rddeploy (1)  wgrib2 (5)
cgal (3)     fdupes (2)     gsl (3)      meson (1)     pdal (2)     rdxadmin (5)  wreport (3)
climetlab (6)  ferret (2)     hat (7)      modulemgr (2)  pdftk (1)    rstudio (4)   wxparaver (3)
cmake (6)    ffmpeg (3)     hdfview (3)  mongodb (3)   pgen (65)    rust (1)      xconv (1)
cmor (5)     flex (1)       idl (2)      mpfr (1)      pifsen (8)   s3cmd (2)     xmars (4)
coda (6)     forge (3)      ifs-git-tools (4)  mpiutils (2)  postgres (1)  seasonal-tools (1)  xthi (1)
comfies (6)  gdal (6)       ifs (61)     mspds (1)     pproc (19)  singularity (1)
cppcheck (3)  gdl (4)        intel-mkl (7)  nccmp (2)     pre-commit (1)  snakemake (4)
cuda (2)     gecko (8)      iris (4)     ncl (1)       prepifs (3)  sqlite (4)

----- /usr/local/apps/modulefiles/lmod/compilers -----
aocc (3)  clang (3)  gcc (10)  intel (7)  nvidia (5)  pgi (1)  rocm-afar (1)

----- /usr/local/apps/modulefiles/lmod/flavours/gnu/8.5 -----
amd-fftw (4)  bufrdc (1)  ecmwf-toolbox (29)  fgsl (2)  hdf5 (5)  hpcx-openmpi (8)  libflame (1)  netcdf4 (5)  openmpi (4)
boost (5)     eclib (4)  fftw (3)           hdf (2)  hdf5 (1)  intel-mpi (7)    nag (1)       openblas (5)  pgen (1)

----- /usr/local/apps/modulefiles/lmod/mpi/gnu/8.5/hpcx/2.9 -----
adios2 (3)  hdf5-parallel (4)  netcdf4-parallel (4)  parmetis (1)  petsc (3)  pnetcdf (2)

----- /usr/local/apps/modulefiles/lmod/flavours/hpcx/2.9 -----
darshan (4)  extrae (3)
```





Software Stack: ECMWF Software

- Some ECMWF software lives in the **ecmwf-toolbox**
 - ecCodes
 - Magics
 - Metview
 - CodesUI
 - ODC
- A single `module load ecmwf-toolbox` to use them all!
- Run `module help ecmwf-toolbox` to get the details of the bundled packages and libraries
- Note that other packages, such as **ecFlow** still have their standalone module:

<https://confluence.ecmwf.int/x/YBhbDg>



Python Support

- Traditional Python 3 central installation
 - + 400 extra Python modules | `module help python3`
- Use your own environment: **venv** | **tykky** | **conda**
 - Python virtual environment `python3 -m venv`
 - `--system-site-packages` option recommended for performance and space savings
 - Conda environments: Use tykky to containerise your stack
 - Better performance and use of the filesystems.

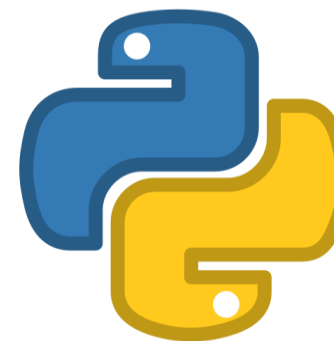
```
module load tykky
```



```
conda-containerize new --mamba --prefix $TYKKY_PATH/mymlenv env.yml
```



```
tykky activate mymlenv
```
 - Only use standard conda when containerised solution does fit your needs.
 - Use of PERM to host your environments is not recommended
 - use HOME for small ones or HPCPERM for bigger needs.



<https://confluence.ecmwf.int/x/YRhBDg>

Container support



- Docker is not supported
- You may use Apptainer if you wish to run containerised workloads.
 - Previously known as Singularity
 - Rootless containers
 - Supports docker and other OCI images
 - BYOE: Bring Your Own Environment.
Develop in your laptop, run in our HPCF!



<https://confluence.ecmwf.int/x/YhhbDg>

```
$ module load apptainer
$ apptainer exec docker://ubuntu:latest cat /etc/os-release
INFO:   Converting OCI blobs to SIF format
INFO:   Starting build...
Copying blob d9d352c11bbd done   |
Copying config bf16bdcff9 done   |
Writing manifest to image destination
2025/06/04 10:35:29  info unpack layer:
sha256:d9d352c11bbd3880007953ed6eec1cbace76898828f3434984a0ca60672fdf5a
INFO:   Creating SIF file...
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```



Interactive sessions

- Limited resources on standard SSH sessions on main login node
- **ecinteractive:** For more demanding interactive workload

```
$ ecinteractive -h
```

```
Usage : /usr/local/bin/ecinteractive [options] [--] [CMD]
```

```
-d|desktop    Submits a vnc job (default is interactive ssh job)
-j|jupyter    Submits a jupyter job (default is interactive ssh job)
-J|jupyterS   Submits a jupyter job with HTTPS support (default is interactive ssh job)
```

More Options:

```
-h|help       Display this message
-v|version    Display script version
-p|platform   Platform (default hpc. Choices: hpc, ecs)
-u|user       ECMWF User (default <your username>)
-A|account    Project account
-c|cpus       Number of CPUs (default 2)
-m|memory     Requested Memory (default 8 GB)
-s|tmpdirsize Requested LOCALSSD/TMPDIR size (default 3 GB)
-g|gpu        Request a GPU (limited availability)
-t|time       wall clock limit (default 12:00:00)
-f|forward    Ports to forward, comma separated (default NONE)
-e|export     Environment variables to export, comma separated (default NONE)
-k|kill       Cancel any running interactive job
-q|query      Check running job
-Q|quiet      Silent mode
-o|output     Output file for the interactive job (default /dev/null)
-x           set -x
```

<https://confluence.ecmwf.int/x/dCc0Dw>



Questions?

