

Urblauna

Urblauna is the UNIL cluster for sensitive depersonalized data.

`stockage-horus` is the name used for the Jura cluster when connecting from the CHUV.

Note: "HORUS" is an acronym for **H**ospital **R**esearch **U**nified data and analytics **S**ervices

Documentation

Associated documentation:

<https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/urblauna-access-and-data-transfer>

<https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/urblauna-guacamole-rdp-issues>

Nearly all the documentation for Curnagl is also applicable - see the [HPC Wiki](#)

The introductory course for using the clusters is available [HERE](#)

The slides for our other courses can be consulted [HERE](#)

These courses are often planned and take place in the Biophore auditorium

Support

Please contact the DCSR via helpdesk@unil.ch and start the mail subject with "DCSR Urblauna"

Do not send mails to dcsr-support - they will be ignored.

Specifications

- 18 compute nodes
- 48 cores / 1 TB memory per node
- 2 nodes with NVidia A100 GPUs
- 1PB /data filesystem

- 75TB SSD based /scratch

Total cores: 864

Total memory: 18TB

Memory to core ratio: 21 GB/core

For those of you have already used Curnagl then things will be very familiar.

If the initial resources are found to be insufficient then more capacity can be easily added.

An [Urblauna](#) looks like:



How to connect

For Jura the connection method is different depending on if one connects from the CHUV or the UNIL networks - for Urblauna it's the same for everyone.

The SSH and Web interfaces can be used simultaneously.

Two Factor Authentication

You should have received a QR code which allows you to setup the 2FA - if you lose your code then let us know and we will generate a new one for you.

SSH

```
% ssh ulambda@u-ssh.dcsr.unil.ch
```

```
(ulambda@u-ssh.dcsr.unil.ch) Password:
```

```
(ulambda@u-ssh.dcsr.unil.ch) Verification code:
```

```
Last login: Wed Jan 18 13:25:46 2023 from 130.223.123.456
```


```
[ulambda@urblauna ~]$
```

The 2FA code is cached for 1 hour in case that you connect again.

X11 Forwarding and SSH tunnels are blocked as is SCP

Web

Go to `u-web.dcsr.unil.ch` and you will be asked for your username and password followed by the 2FA code:

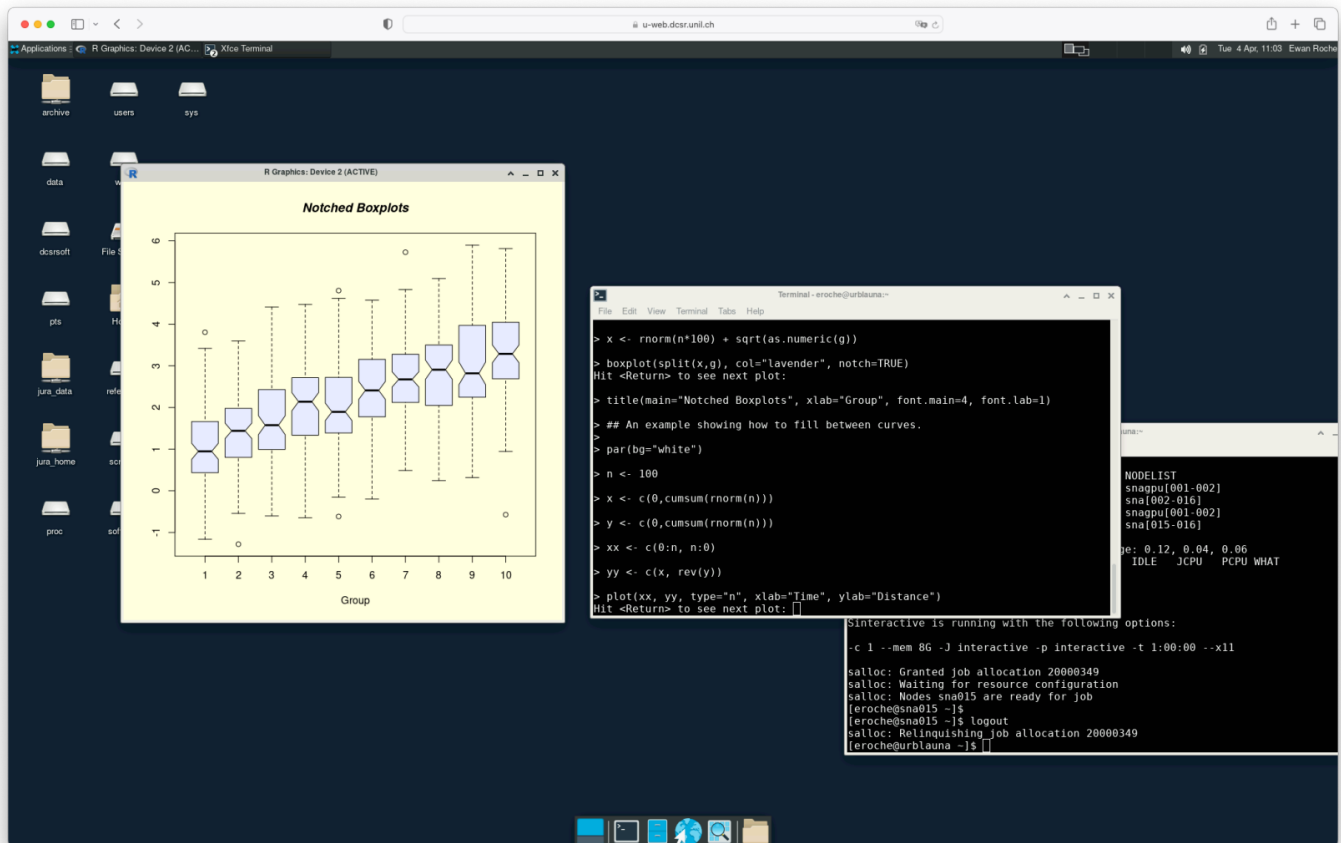


UNIL | Université de Lausanne

URBLAUNA REMOTE DESKTOP

This will send you to a web based graphical desktop.

Note than until now the CHUV users did not have this as a connection option.



Data Transfer

The principle method to get data in/out of Urblauna is using the SFTP protocol

On Urblauna your `/scratch/<username>` space is used as the buffer when transferring data.

```
% sftp ulambda@u-sftp.dcsr.unil.ch
(ulambda@u-sftp.dcsr.unil.ch) Password:
(ulambda@u-sftp.dcsr.unil.ch) Verification code:
Connected to u-sftp.dcsr.unil.ch.

sftp> pwd
Remote working directory: /ulambda

sftp> put mydata.tar.gz
Uploading mydata.tar.gz to /ulambda/mydata.tar.gz
```

The file will then be visible from urblauna at `/scratch/ulambda/mydata.tar.gz`

For graphical clients such as Filezilla you need to use the **interactive** login type so as to be able to enter the 2FA code.

Direct data transfer from CHUV

There is also an SFTP endpoint `u-archive.dcsr.unil.ch` that will allow transfer to the `/archive` filesystem without 2FA from **specific IP** addresses at the CHUV.

This service will be what `stockage-horus` was originally supposed to be!

This is on request and must be validated by the CHUV security team.

What's new

There are a number of changes between Jura and Urblauna that you need to be aware of:

CPUs

The nodes each have two AMD Zen3 CPUs with 24 cores for a total of 48 cores per node.

In your jobs please ask for the following core counts:

- 1
- 2
- 4
- 8
- 12
- 24
- 48

Do not ask for core counts like **20 / 32 / 40** as this makes no sense given the underlying architecture. We recommend running scaling tests to find the optimal level of parallelism for multi-threaded codes.

Unlike for Jura all the CPUs are identical so all nodes will provide the same performance.

GPUs

There are two GPU equipped nodes in Urblauna and each A100 card has been partitioned so as to provide a total of 8 GPUs with 20GB of memory.

To request a GPU use the `--gres` Slurm directive

```
#SBATCH --gres gpu:1
```

Or interactively with `Sinteractive` and the `-G` option

\$ Sinteractive -G 1

Sinteractive is running with the following options:

--gres=gpu:1 -c 1 --mem 8G -J interactive -p interactive -t 1:00:00

salloc: Granted job allocation 20000394

salloc: Waiting for resource configuration

salloc: Nodes snagpu001 are ready for job

\$ nvidia-smi

..

+-----+															
MIG devices:															
+-----+-----+-----+-----+															
GPU GI CI MIG				Memory-Usage				Vol		Shared					
ID ID Dev				BAR1-Usage SM				Unc CE ENC DEC		OFA JPG					
				ECC											
=====+=====+=====+=====															
=====															
0 1 0 0				19MiB / 19968MiB				42 0		3 0 2 0 0					
				0MiB / 32767MiB											
+-----+-----+-----+-----+															
+-----+															
Processes:															
GPU GI CI				PID Type Process name				GPU Memory							
ID ID				Usage											
=====															
=====															
No running processes found															
+-----+															

MPI

This is now possible... Ask us for more details if needed.

/data

The `/data` filesystem is structured in the same way as on Curnagl

```
/ data / FAC / FACULTY / INSTITUTE / PI / PROJECT
```

This space is on reliable storage but there are no backups or snapshots.

The initial quotas are the same as on Jura - if you wish to increase the limit then just ask us. With 1PB available all reasonable requests will be accepted.

/scratch

Unlike on Jura `/scratch` is now organised per user as on Curnagl and as it is considered as temporary space there is no fee associated.

There are no quotas but in case of the utilisation being greater than 90% then files older than 2 weeks will be removed automatically.

/users

The `/users` home directory filesystem is also new - the Jura home directories can be accessed in read-only at `/jura_home`.

/work

The Curnagl `/work` filesystem is visible in read-only from inside Urblauna. This is very useful for being able to install software on an Internet connected system.

/reference

This is intended to host widely used datasets

The `/db` set of biological databases can be found at `/reference/bio_db/`

/archive

This is an HSM (Hierarchical Storage Management system) meaning that any files written are copied on tape in two copies, after some time the file content is erased from disk and a pointer to the file on tape remains. If you open a file which content is not on disk any more the tape cartridge has to be loaded in a drive, spooled to the right place and then transferred to disk. It is only for cold data. If you have to retrieve more than 1000 files please send us a ticket at helpdesk@unil.ch with subject "DCSR Urblauna archive retrieve" and the directory path. It has the same organization as `/data`

```
/ archive / FAC / FACULTY / INSTITUTE / PI / PROJECT
```

The DCSR software stack

This is now the default stack and is identical to Curnagl. It is still possible to use the old Vital-IT /software but this is deprecated and no support can be provided.

For how to do this see the documentation at [Old software stack](#)

Note: The version of the DCSR software stack currently available on Jura (*source /dcsrsoft/spack/bin/setup_dcsrsoft*) is one release behind that of Curnagl/Urblauna. To have the same stack you can use the `dcsrsoft use old` command:

```
[ulambda@urblauna ~]$ dcsrsoft use old  
Switching to the old software stack
```

There's lots of information on how to use this in our [introductory course](#)

Installing your own software

We encourage you to ask for a project on Curnagl (HPC normal data) which will allow you to install tools and then be able to use them directly inside Urblauna.

See [the documentation](#) for further details

For those who use Conda don't forget to make sure that all the directories are in your project /work space

<https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/using-conda-and-anaconda>

```
nano .condarc  
  
pkgs_dirs:  
- /work/path/to/my/project/space  
  
envs_dirs:  
- /work/path/to/my/project/space
```

For R packages it's easy to set an alternative library location:

```
echo 'R_LIBS_USER=/work/path/to/project/Rlib' > ~/.Renvron
```

This will need to be run on both Curnagl and Urblauna and will allow you to install packages when connected to the internet and run them inside the air gapped environment.

Révision #5

Créé 6 juin 2025 14:10:03 par Roberto Fabbretti

Mis à jour 7 juin 2025 11:05:41 par Johann Fleury