Filetransfer from the cluster

filetransfer.dcsr.unil.ch

https://filetransfer.dcsr.unil.ch is a service provided by the DCSR to allow you to transfer files to and from external collaborators.

This is an alternative to SWITCHFileSender and the space available is 6TB with a maximum per user limit of 4TB - this space is shared between all users so it is unlikely that you will be able to transfer 4TB of data at once.

The filetransfer service is based on LiquidFiles and the user guide is available at

https://man.liquidfiles.com/userguide.html

In order to transfer files to and from the DCSR clusters without using the web browser it is also possible to the API REST as explained below

Configuring the service

First you need to connect to the web interface at https://filetransfer.dcsr.unil.ch and connect using your UNIL username (e.g. ulambda for Ursula Lambda) and password. This is not your EduID password but rather the one you use to connect to the clusters.

Once connected go to settings (the cog symbol in the top right corner) then the API tab

Account Settings

| Basic Settings | Two-Factor Authentication | Previous Recipients | Contacts | Personal Filedrop | API |

API Key

The API key is used for plugins such as the Outlook plugin and can be used for custom integration. You only need this if an admin tells you to.

API Key   9MUavQeF4uRAy049lHdCtg   [📋 Copy to Clipboard] [↺ Reset API key]

The API key is how you authenticate from the clusters and this secret should never be shared. It can be reset via the yellow button.

Transferring files from the cluster

For example to upload a file and create a file link

```
curl  -X POST -H "Accept: application/json"  -H "Content-Type: application/zip"\
     --user 9MUQeF5nG899lHdCtg:x \
     --data-binary @myfile.data \
     "https://filetransfer.dcsr.unil.ch/message/attachments/upload?filename=myfile.data"
```

Make sure to put `--user APIKEY:x` as a parameter.

You can then connect to the web interface from you workstation to manage the files and send messages as required.

As preparing and uploading files can take a while we recommend that this is performed in a tmux session which means that even if your connection to the cluster is lost the process continues and you can reconnect.

Transferring large files

If using a single file upload doesn't work and it is not possible to split the data into multiple smaller files then the following information may be useful

Staging the files

We recommend that you create TAR files containing the data you wish to transfer and stage this in your /scratch space. Depending on the data type it can be useful to compress it first.

```
$ cd /scratch/ulambda
$ mkdir mytransfer
$ cd mytransfer
$ tar -cvf mydata.tar /work/path/to/my/data
```

Then calculate the checksum of the file to be transfered

```
$ sha256sum mydata.tar
7aac249b9ec0835361f44c84921a194e587a38daecadf302e9dec44386c9fb36  mydata.tar
```

Split the file and transfer chunks

Whilst it might be possible to transfer huge files in one upload, it isn't recommended and above ~100GB we recommend that you follow the procedure given below.

Split the file into chunks

```
$ split --verbose -d -a4 -b1G mydata.tar
creating file 'x0000'
creating file 'x0001'
```

```
creating file 'x0002'

creating file 'x0003'

..

..

creating file 'x0102'
```

In the staging directory this will create files of exactly 1GB in size- here Usrula's file is 102.5 GB so there are 103 chunks

Use a loop and the attach_chunk command

First we need to know how many files there are

```
$ ls x* | wc -l
103
```

This is because we need to tell the service how many bits the file has been split into so it knows when the upload is complete.

Now we use our API key as a parameter for `--user` option and use the following bash loop (this can also be put in a script).

```
CHUNKS=103

MYFILE=mydata.tar

for a in $(seq -w 0 $CHUNKS)

do

    curl  -X POST -H "Accept: application/json"  -H "Content-Type: application/zip"\

        --user 9MUQeF5nG899lHdCtg:x \

        --data-binary @x0$a \


"https://filetransfer.dcsr.unil.ch/message/attachments/upload?filename=$MYFILE&chunk=$a&chunks
=$CHUNKS"

done
```

Once all the chunks are uploaded the file will be assembled/processed and after a short while it will be visible in the web interface.

Here we see a previously uploaded file of 304 GB called my file.ffdata

Attached files

myfile.ffdata	304 GB	🗑 Delete

⬇ **Drop Files Here**

1 file (304 GB)

+ Add Files... ▾

Cleaning up

Once the file is uploaded please don't forget to clean up the TAR file and the chunks.

```
$ cd /scratch/ulambda/mytransfer
$ rm *
$ cd ..
$ rmdir mytransfer
```