Tools for high speed data transfer

Aleksandrs Gutcaits, 2020-10-23 - Riga

Genomics Research and Data Transfer

- Genomics research today relies on massive analysis of extra-large data. Currently, the size of a single genome's data set is around 100GB – and it's estimated to be grown tens or even hundreds of terabytes soon.
- Sequence analysis: Homology searches, Genome comparisons, Genome-wide analyses
- Sequencing data are expected to increase more rapidly
- Requires large databases such as DNA and protein sequence. Sharing and managing biological databases become more and more difficult and intractable
- Researchers needs everyday tools for big data transfer and manipulation

In this presentation possible usage such tools as SCP, SFTP, RSYNC and Globus for data transfer and manipulation will be discussed with measurement of current transfer speed.

SCP and SFTP Data Transfer Tools

- SCP stands for **Secure Copy Protocol**. Protocol allows transferring files between the local host and a remote host or between two remote hosts.
- In addition to **file transfer**, SCP also supports encryption and authentication features. This protocol uses Secure Shell (SSH) protocol.
- SCP command-line client does not allow resuming an interrupted file transfer.

- SFTP stands for **Secure File Transfer Protocol**. It allows accessing and transferring files, managing the files over a reliable data stream.
- In addition to file transfers, SFTP allows performing tasks such as **creating directories, delete directories, delete files etc**.
- Unlike in SCP, SFTP sends an acknowledgement for every packet. Therefore, **SFTP is slower than SCP.**
- SFTP command-line client allows resuming an interrupted file transfer with the –a option of the 'get' command.

The main difference between SCP and SFTP is that SCP is a protocol that allows transferring files securely from a local host to a remote host while SFTP is a protocol that allows file accessing, transferring, and management over a reliable data stream which is faster than SCP.

SCP Command Options

The scp command syntax take the following form:

scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2

scp provides a number of options that control every aspect of its behavior. The most widely used options are:

-P - Specifies the remote host ssh port.
-p - Preserves files modification and access times.nds options
-q - Use this option if you want to suppress the progress meter and non-error messages.
-C - This option forces scp to compresses the data as it is sent to the destination machine.
-r - This option tells scp to copy directories recursively
-v - Verbose mode. Prints debugging messages about their progress.

SCP between RTU - LU

```
[gutzeit@ui-1 gutzeit]$ scp -rv gutzeit_home_rtu_10GB.file  cms001@10.10.115.100:/tr2home/cms001/
Executing: program /usr/bin/ssh host 10.10.115.100, user cms001, command scp -v -r -t /tr2home/cms001/
```

```
Sending file modes: C0644 10000000000 gutzeit_home_rtu_10GB.file
Sink: C0644 10000000000 gutzeit_home_rtu_10GB.file
gutzeit_home_rtu_10GB.file                                    10% 1003MB 108.4MB/s   01:18 ETAd
```

```
debug1: ssh_set_newkeys: rekeying after 16737 input blocks (168364 bytes total)
debug1: rekey after 134217728 blocks
gutzeit_home_rtu_10GB.file                                    20% 1978MB 109.0MB/s   01:09 ETAd
```

```
gutzeit_home_rtu_10GB.file                                   100% 9537MB  90.8MB/s   01:45
```

```
Transferred: sent 10006431952, received 1606748 bytes, in 105.7 seconds
Bytes per second: sent 94667436.0, received 15200.9
```

SCP BetweenTwo Remote Systems (RTU – LU)

```
$ scp user1@host1.com:/files/file.txt     user2@host2.com:/files
```

```
[cms001@tr2sl-1 ~]$ scp –rv -C gutzeit@10.10.112.76:/home/groups/globus/gutzeit/gutzeit_home_rtu_10GB.file
                    cms001@10.10.115.100:/tr2home/cms001/
…
gutzeit_home_rtu_10GB.file                                    100% 9537MB  94.4MB/s   01:41
Transferred: sent 10006740984, received 1613328 bytes, in 102.1 seconds
Bytes per second: sent 97980178.2, received 15796.8
```

```
[cms001@tr2sl-1 ~]$ scp -r gutzeit@10.10.112.76:/home/groups/globus/gutzeit/gutzeit_home_rtu_10GB.file
                    cms001@10.10.115.100:/tr2home/cms001/
gutzeit@10.10.112.76’s     password:
cms001@10.10.115.100's password:
gutzeit_home_rtu_10GB.file                                    100% 9537MB  84.4MB/s   01:53
```

SCP between RTU - BMC

```
[gutzeit@ui-1 gutzeit]$ scp  -C gutzeit_home_rtu_10GB.file gutzeit@10.245.1.149:/home/gutzeit/
gutzeit_home_rtu_10GB.file                    100% 9537MB 179.6MB/s   00:53
```

```
gutzeit@PC2:~$ scp -rv  -C gutzeit_PC2_10GB.file gutzeit@85.254.226.76:/home/groups/globus/gutzeit/
gutzeit_PC2_10GB.file                         100% 9537MB 160.8MB/s   00:59
```

```
Transferred: sent 26857492, received 1786260 bytes, in 60.0 seconds
Bytes per second: sent 447503.6, received 29762.9
```

Using rsync for Files Transfer

Advantages and features of Rsync command

Rsync (Remote Sync) is a most commonly used command for **copying** and **synchronizing** files and directories **remotely** as well as locally in **Linux**/**Unix** systems.

- It efficiently copies and sync files to or from a remote system.
- Supports copying links, devices, owners, groups and permissions.
- It's faster than **scp** (**Secure Copy**) because **rsync** uses remote-update protocol which allows to **transfer just the differences between two sets of files**. First time, it copies the whole content of a file or a directory from source to destination but **from next time, it copies only the changed blocks and bytes to the destination.**
- Rsync consumes **less bandwidth** as it uses compression and decompression method while sending and receiving data both ends.
- **rsync** command you using for copy and synchronize your data remotely and locally across directories, across disks and networks, perform data backups and mirroring between two Linux machines.

Main rsync Command Options

```
# rsync options source destination
```

- **-A,** --acls preserve ACLs (implies --perms)
- **-v** : verbose
- **-r** : copies data recursively (but don't preserve timestamps and permission while transferring data
- **-a** : archive mode, archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps
- **-z** : compress file data
- **-h** : human-readable, output numbers in a human-readable format
- **-X**, --xattrs preserve extended attributes
- -d, --dirs. transfer directories without recursing
- --progress show progress during transfer

rsync Tranfer Between RTU - LU

```
[gutzeit@ui-1 gutzeit]$ rsync -aAXvz --progress rsync_rtu/  cms001@10.10.115.100:/tr2home/cms001/
gutzeit_home_rtu_10GB_01.file
 10,000,000,000 100%  135.66MB/s    0:01:10 (xfr#1, to-chk=3/5)
gutzeit_home_rtu_10GB_02.file
 10,000,000,000 100%  131.39MB/s    0:01:12 (xfr#2, to-chk=2/5)
gutzeit_home_rtu_10GB_03.file
 10,000,000,000 100%  113.04MB/s    0:01:24 (xfr#3, to-chk=1/5)
gutzeit_home_rtu_10GB_04.file
 10,000,000,000 100%   94.69MB/s    0:01:40 (xfr#4, to-chk=0/5)
```

RTU => LU

```
[cms001@hpc ~]$ rsync -aAXvz --progress rsync_lu gutzeit@10.10.112.76:/home/groups/globus/gutzeit/
rsync_lu/gutzeit_home_lu_10GB_01.file
 10,000,000,000 100%   84.16MB/s    0:01:53 (xfr#1, to-chk=3/5)
rsync_lu/gutzeit_home_lu_10GB_02.file
 10,000,000,000 100%   85.05MB/s    0:01:52 (xfr#2, to-chk=2/5)
rsync_lu/gutzeit_home_lu_10GB_03.file
 10,000,000,000 100%   84.95MB/s    0:01:52 (xfr#3, to-chk=1/5)
rsync_lu/gutzeit_home_lu_10GB_04.file
 10,000,000,000 100%   85.27MB/s    0:01:51 (xfr#4, to-chk=0/5)
```

LU => RTU

Globus Connect Personal and Server

- Globus augments in secure copy (scp/sftp) requests by automating reliable large data transfers, by resuming failed transfers, encrypts transfers and by simplifying the implementation of high-performance transfers between computing centers.

Globus Connect Personal

Creates a Globus endpoint on your laptop or other personal computer and allows you to transfer and share files, even if you don't have administrative privileges on your machine. Globus Connect Personal is available for Mac OS X, Windows, and Linux operating systems.

Globus Connect Server

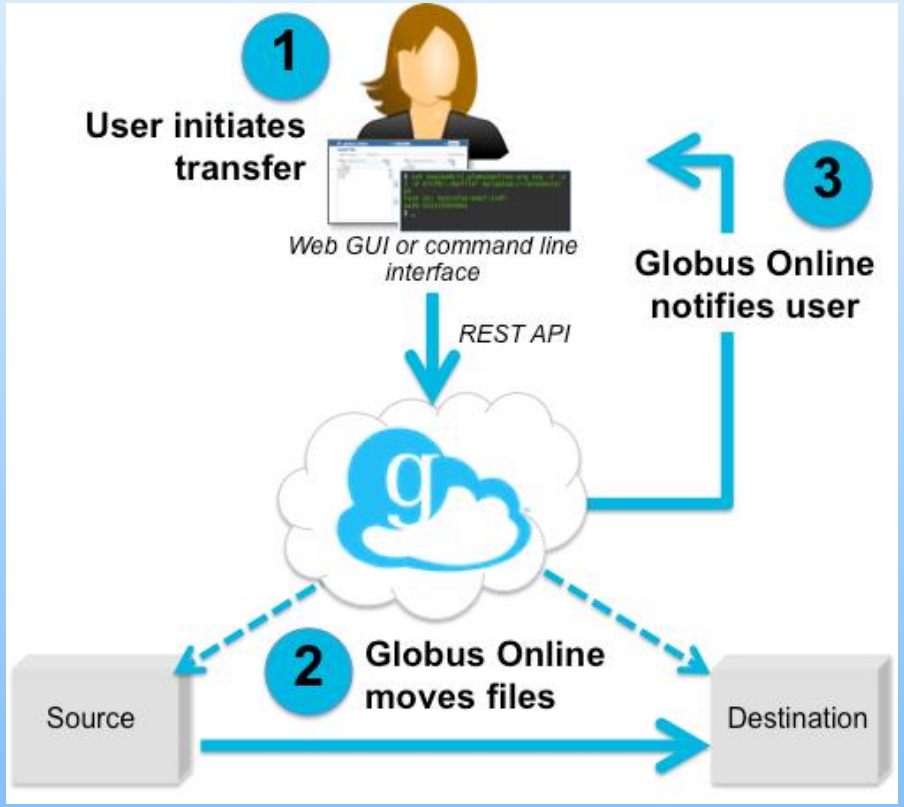
Creates a Globus endpoint on multi-user systems such as a lab servers, campus research computing clusters, and other high-performance computing or storage resources. Globus Connect Server is available for all POSIX-compliant filesystems, and many object stores and tape archives.

https://www.globus.org/

Globus Connect Main Features

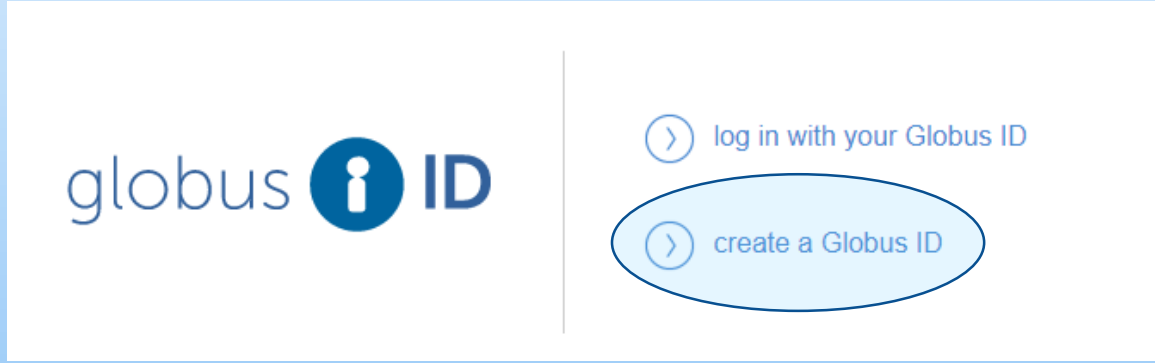
Move, sync, share files
– Easy “fire-and-forget” transfers
– Share with any Globus user or group
– Automatic fault recovery & high performance
– Across multiple security domains
– Web, command line, and REST interfaces

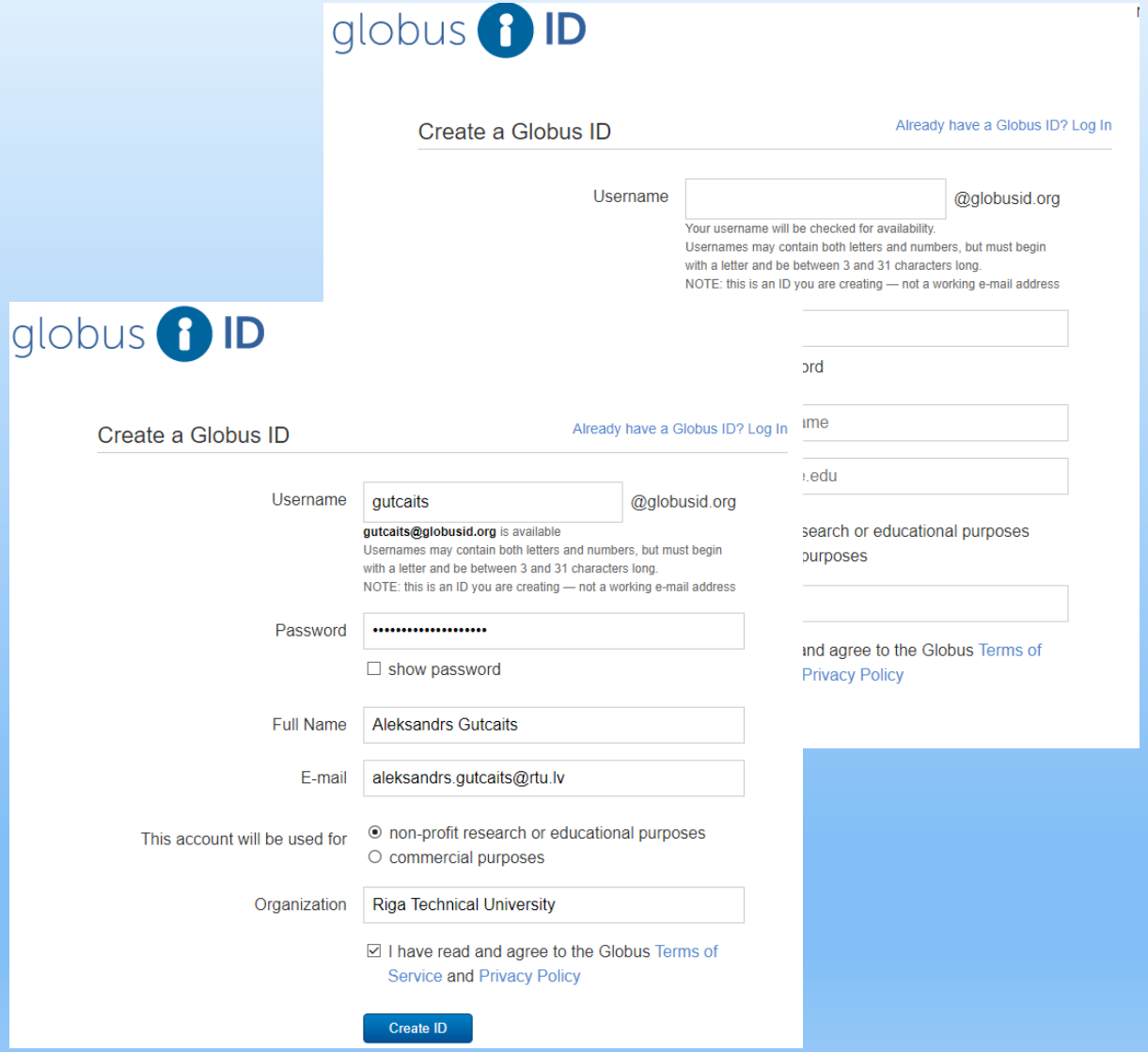
• **Minimize IT costs**
– Software as a Service (SaaS)
• No client software installation
• New features automatically available
– Consolidated support & troubleshooting
– Simple endpoint installation with Globus Connect and GridFTP



https://www.globus.org/

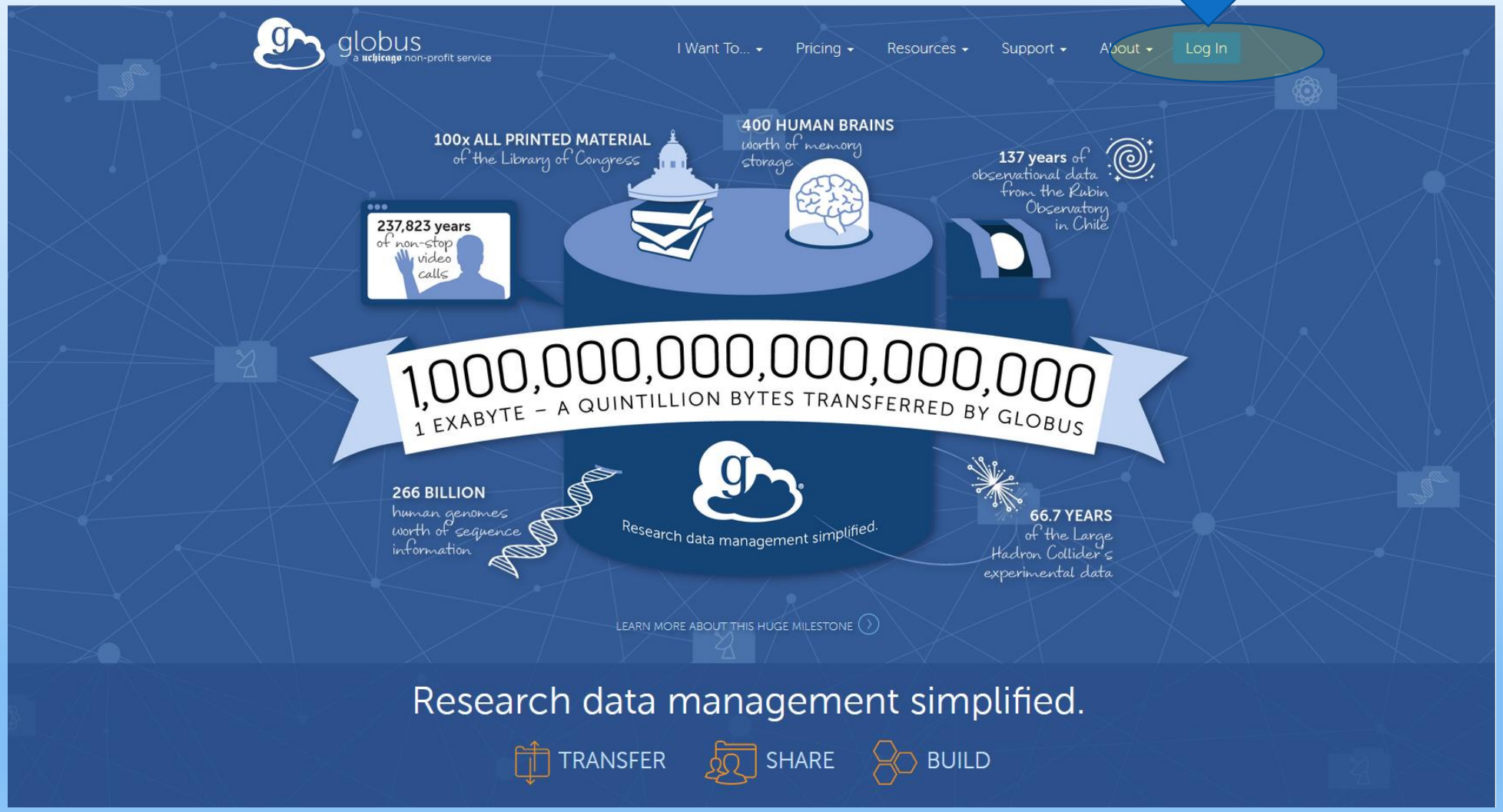
Globus Connect Account Creation





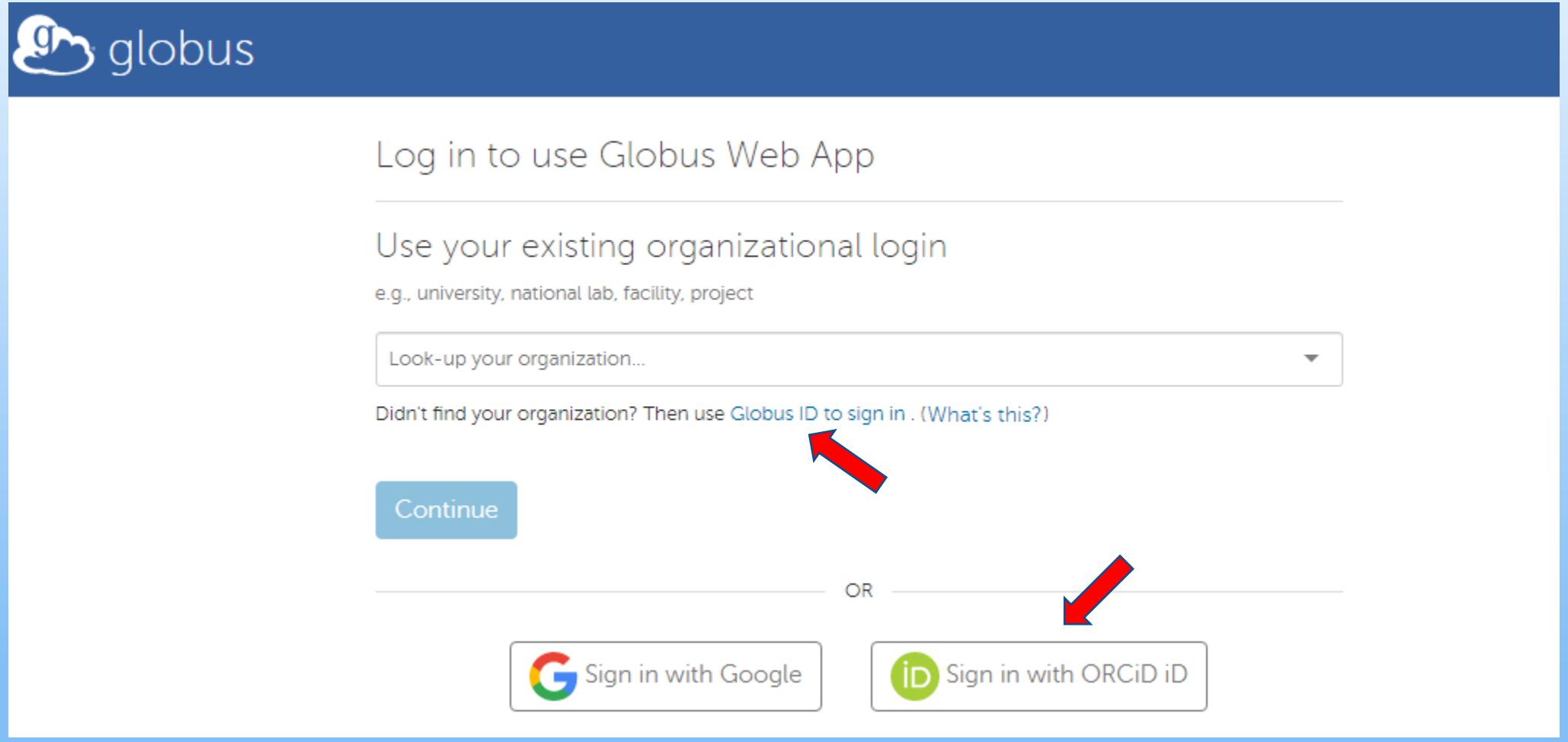
https://www.globusid.org/

Globus Connect Main WEB Interface

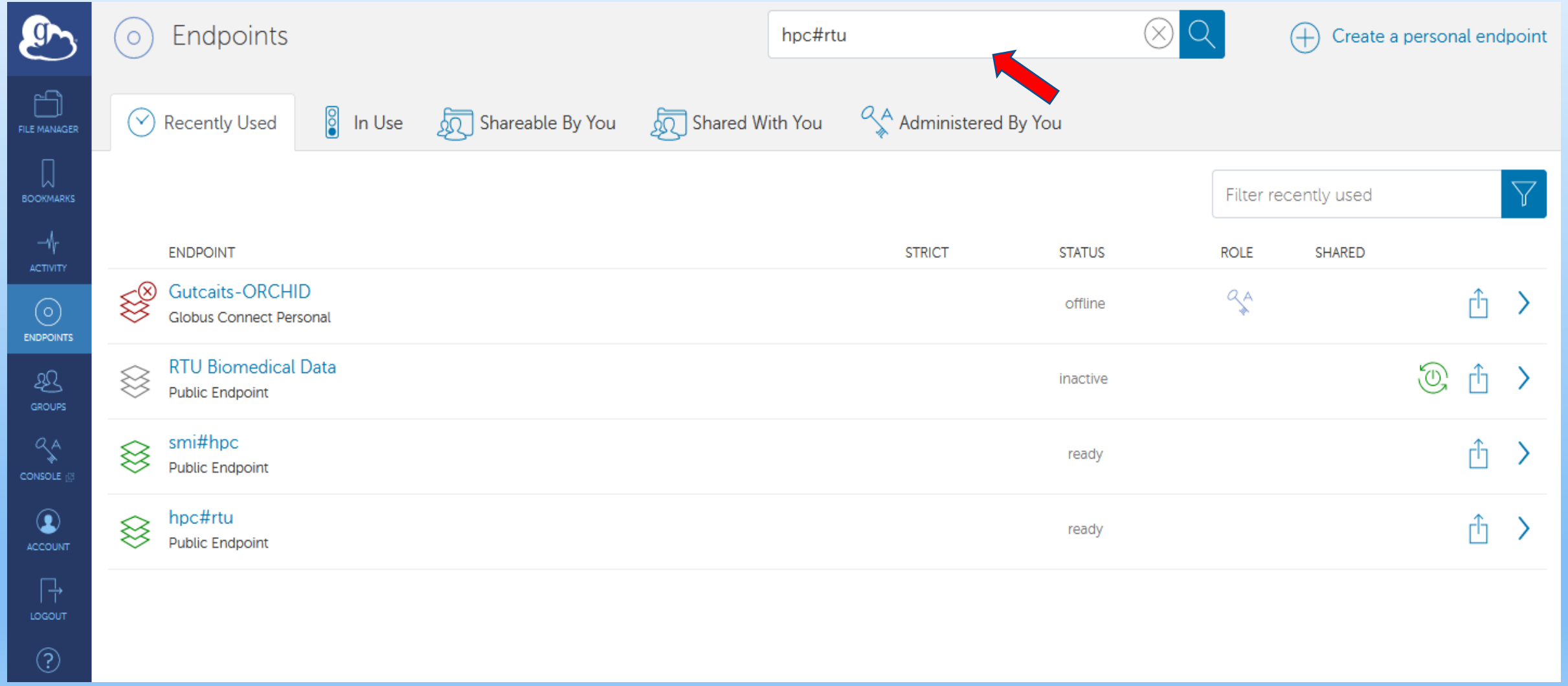


https://www.globus.org/

Globus Connect Login

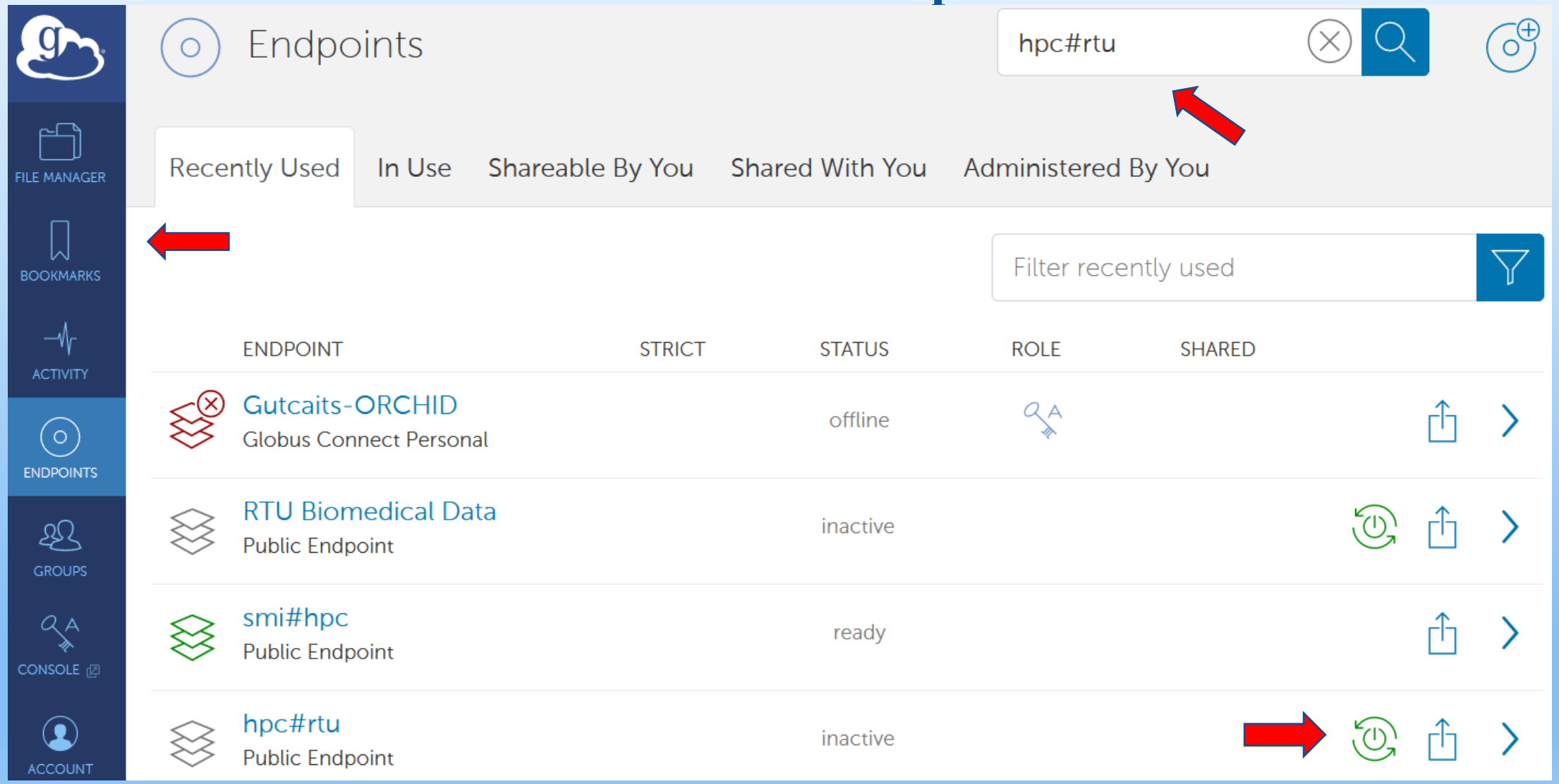


Globus Connect Endpoints



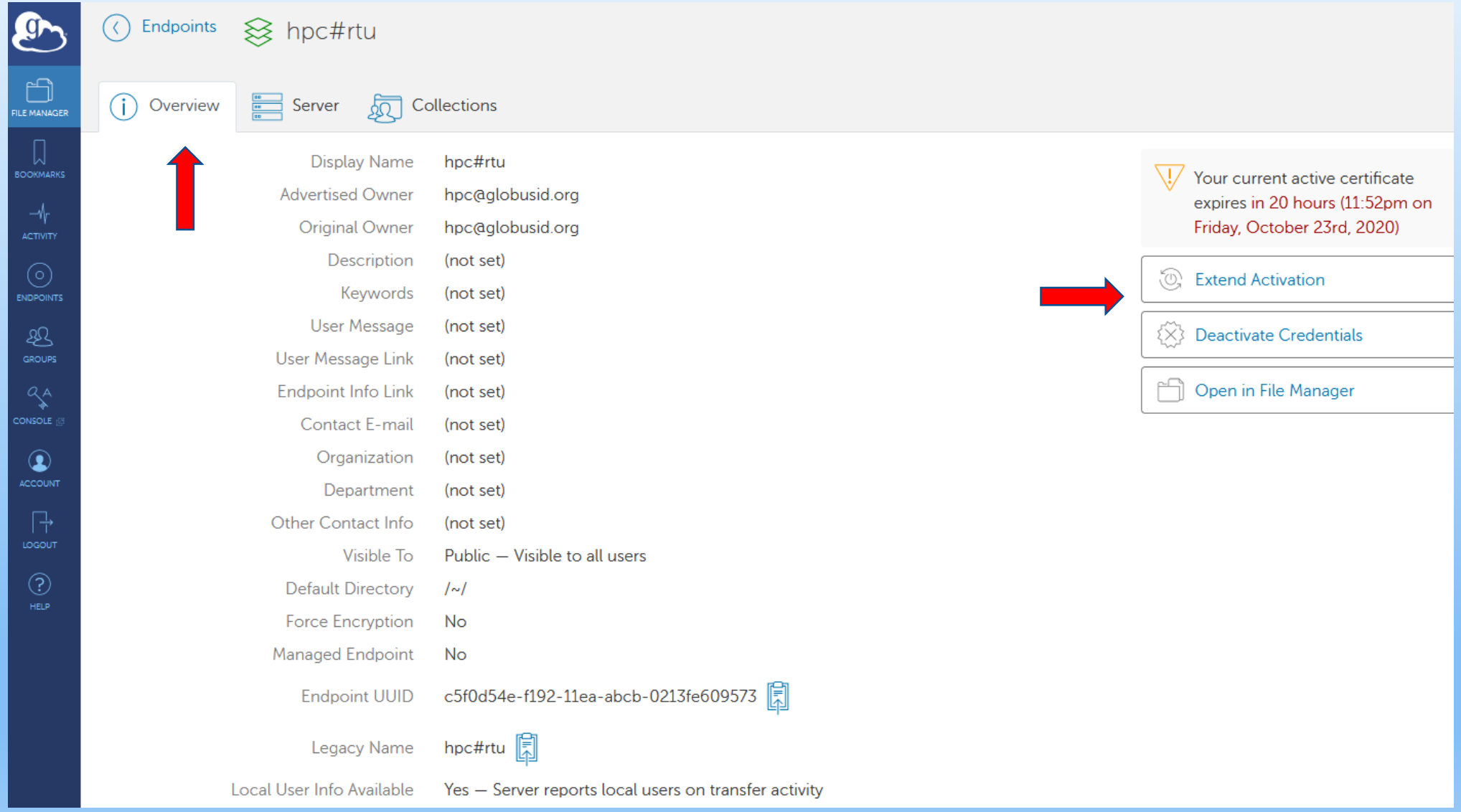
https://www.globus.org/

Globus Connect Endpoints



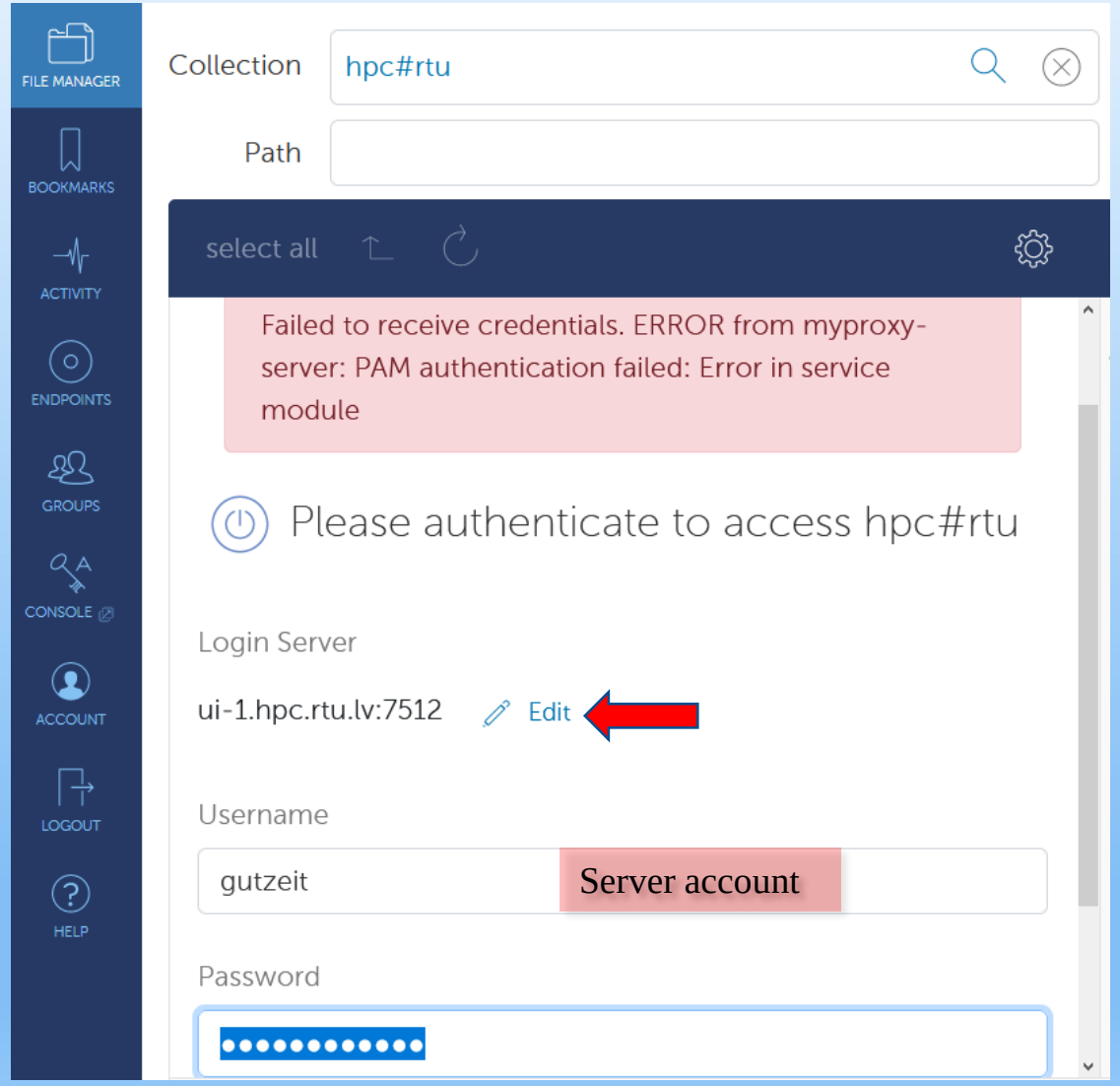
https://www.globus.org/

Globus Connect Endpoint



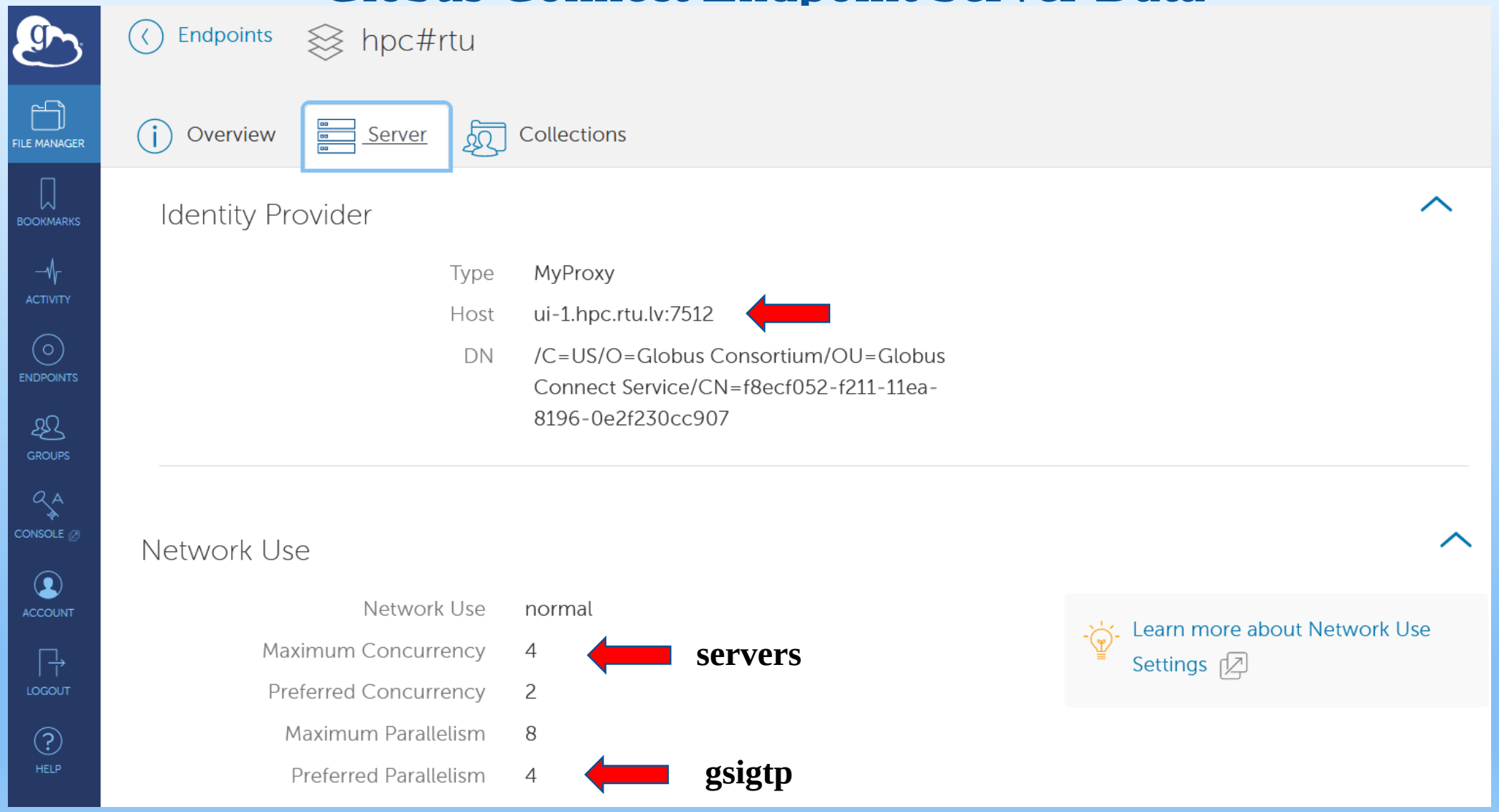
https://www.globus.org/

Globus Connect Endpoint Authenication



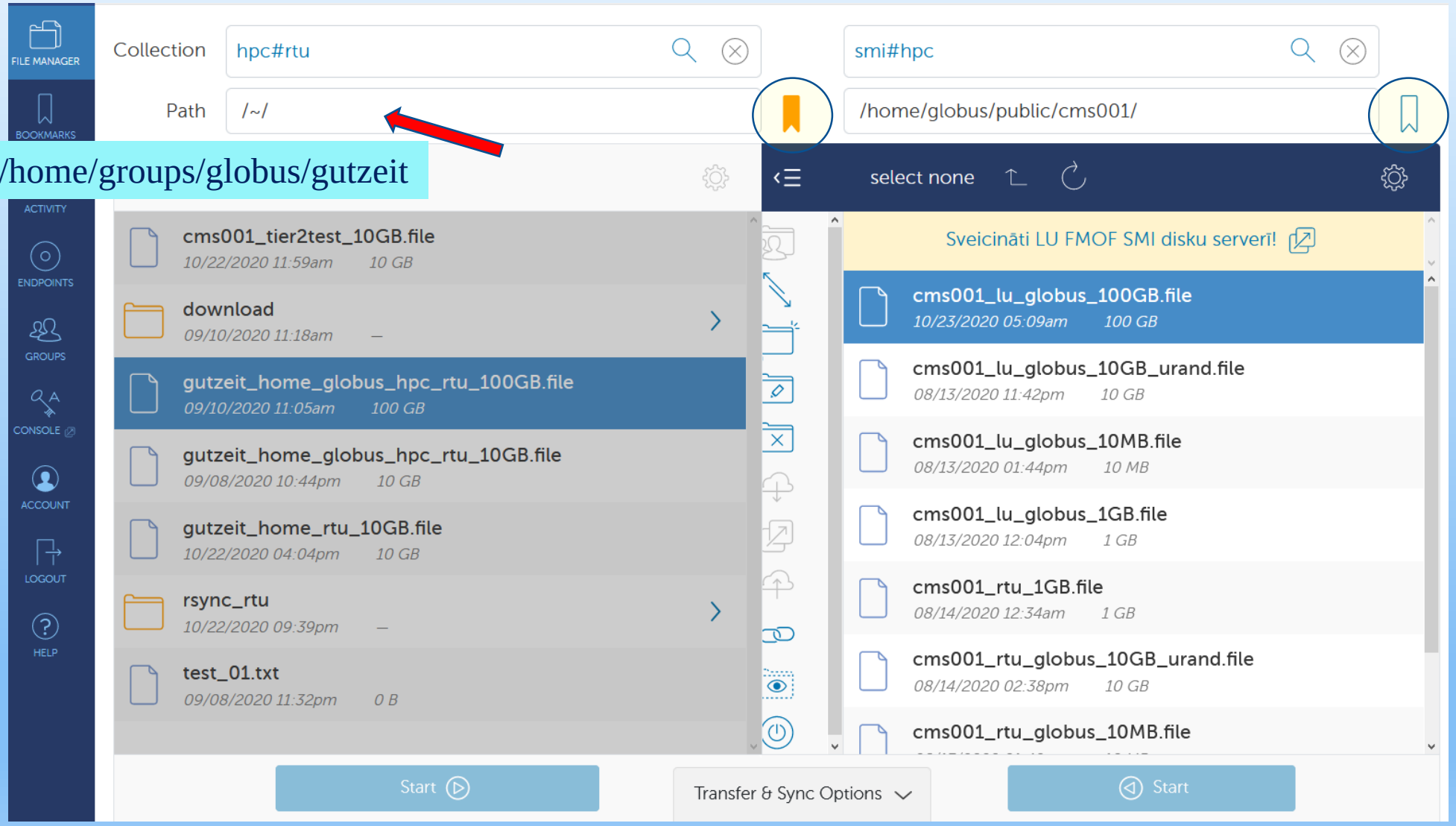
https://www.globus.org/

Globus Connect Endpoint Server Data



https://www.globus.org/

Globus Connect Files Transfer RTU=>LU



$HOME = /home/groups/globus/gutzeit

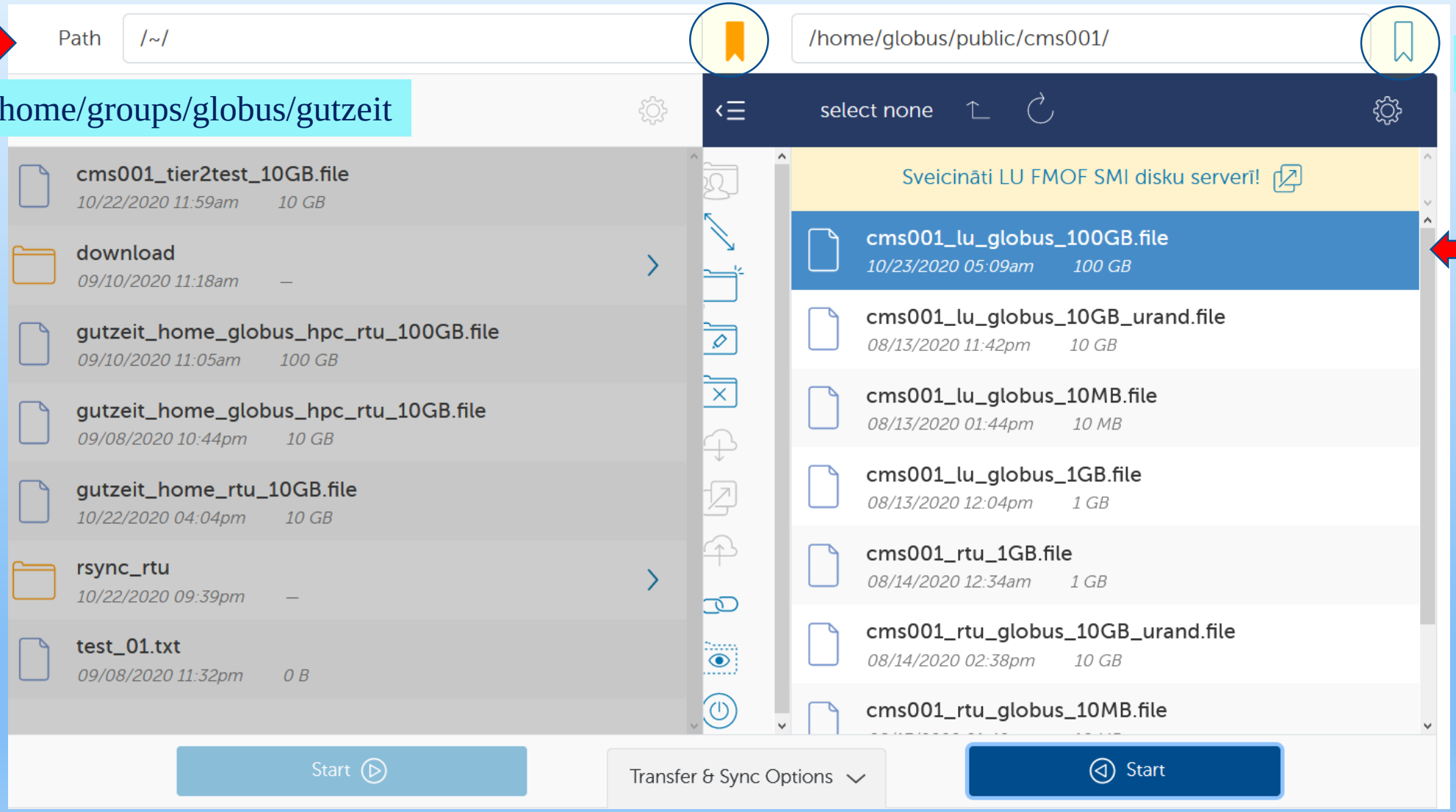
Bookmarks

https://www.globus.org/

Globus Connect Files Transfer LU=>RTU

$HOME = /home/groups/globus/gutzeit

Bookmarks



https://www.globus.org/

Globus Connect File Transfer RTU-LU

RTU => LU, 100GB file

[cms001@hpc ~]$ nload -u M
ens3f0.2020 [10.10.115.100] (6/11):

```
Curr: 55.42 MByte/s
Avg: 221.66 MByte/s
Min: 0.00 MByte/s
Max: 649.81 MByte/s
Ttl: 705.06 GByte
```

Globus overview

1	Files
0	Directories
100 GB	Bytes Transferred
108.45 MB/s	Effective Speed
0	Skipped

LU => RTU, 100GB file

[gutzeit@ui-1 gutzeit]$ nload –u M
eth1.2020 [10.10.112.76] (3/10)

```
Curr: 142.05 MByte/s
Avg: 138.01 MByte/s
Min: 0.00 MByte/s
Max: 347.61 MByte/s
Ttl: 131.54 GByte
```

Globus overview

1	Files
0	Directories
100 GB	Bytes Transferred
91.66 MB/s	Effective Speed
0	Skipped

•.

Globus Connect Personal

Globus Connect Personal turns your laptop or other personal computer into a Globus endpoint With Globus Connect Personal you can share and transfer files to/from a local machine—campus server, desktop computer or laptop - even if it's behind a firewall and you don't have administrator privileges.

- Dramatically increases data transfer speeds over scp and other transfer tools.
- Automatically suspends transfers when computer sleeps and resumes when turned on.
- Installs in seconds using native operating system install packages

https://www.globus.org/globus-connect-personal

ARCHIVE

iperf3 testing on PC2 (BMC)

```
gutzeit@PC2:~$ iperf3 -s -f g
```

```
Server listening on 5201
Accepted connection from 10.245.1.146, port 55300   ===>    rtu ui-1
[  5] local 10.245.1.149 port 5201 connected to 10.245.1.146 port 55302
[  5]   0.00-10.03  sec      11.0 GBytes      9.41 Gbits/sec                  receiver
```

```
Accepted connection from 10.245.1.146, port 55318 ===>     rtu ui-1
[  5] local 10.245.1.149 port 5201 connected to 10.245.1.146 port 55320
[  5]   0.00-10.03  sec      11.0 GBytes     1.10 GBytes/sec                  receiver
```

SCP from tier2test - RTU HPC

```
[cms001@tr2sl-1 ~]$ scp -rv /tr2home/cms001/cms001_tier2test_10GB.filegutzeit@85.254.226.77:/home/groups/globus/gutzeit/
Executing: program /usr/bin/ssh host 85.254.226.77, user gutzeit, command scp -v -r -t /home/groups/globus/gutzeit/
```

```
Sending file modes: C0644 10000000000 cms001_tier2test_10GB.file
Sink: C0644 10000000000 cms001_tier2test_10GB.file
cms001_tier2test_10GB.file                                    10% 1018MB  92.1MB/s   01:32 ETAd
```

```
debug1: ssh_set_newkeys: rekeying after 19118 input blocks (192108 bytes total)
debug1: rekey after 134217728 blocks
cms001_tier2test_10GB.file                                    21% 2031MB  92.1MB/s   01:21 ETAd
```

```
cms001_tier2test_10GB.file                                   100% 9537MB 105.1MB/s   01:30
```

```
Transferred: sent 10012220312, received 1770152 bytes, in 104.8 seconds
Bytes per second: sent 95576954.2, received 16897.9
```

SCP from tier2test to RTU HPC

```
[cms001@tr2sl-1 ~]$ scp -rv /tr2home/cms001/cms001_tier2test_10GB.file gutzeit@85.254.226.77:/home/groups/globus/gutzeit/
Executing: program /usr/bin/ssh host 85.254.226.77, user gutzeit, command scp -v -r -t /home/groups/globus/gutzeit/
```

```
Sending file modes: C0644 10000000000 cms001_tier2test_10GB.file
Sink: C0644 10000000000 cms001_tier2test_10GB.file
cms001_tier2test_10GB.file                                          10% 1018MB  92.1MB/s   01:32 ETAd
```

```
cms001_tier2test_10GB.file                                         100% 9537MB  91.7MB/s   01:44
```

```
Transferred: sent 10012220312, received 1770152 bytes, in 104.8 seconds
Bytes per second: sent 95576954.2, received 16897.9
```

```
debug1: ssh_set_newkeys: rekeying after 19118 input blocks (192108 bytes total)
debug1: rekey after 134217728 blocks
cms001_tier2test_10GB.file                                          21% 2031MB  92.1MB/s   01:21
```

1

2

3