Storage and Data Management at BioHPC

Cornell Bioinformatics Facility workshop

Nov 15, 2021

Outline

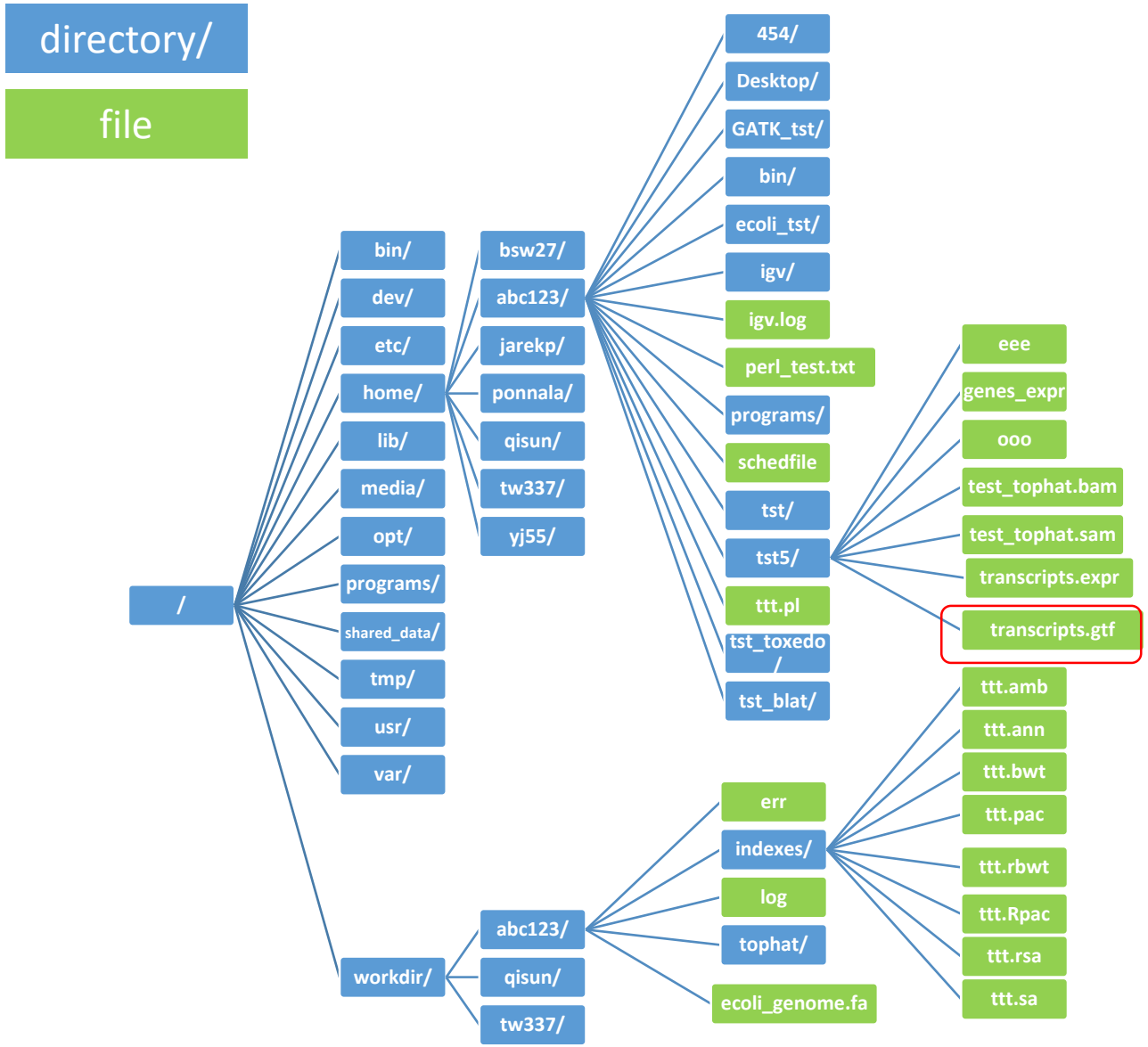
Storage at BioHPC – an overview

Importance of backup, strategies and locations

Data sharing with outside collaborators

Data sharing between BioHPC users

Files and directories on BioHPC Linux machines



Directories (aka folders) allow to organize data, speed up access

Referring to files:

Full path:

`/home/abc123/tst5/transcripts.gtf`

Relative path (i.e., relative to **`/home/abc123`**)

`tst5/transcripts.gtf`

Relative path (i.e., relative to **`/home/abc123/tst5`**)

`transcripts.gtf`

1 - 10 - 700 TB

Storage at BioHPC

/workdir
/SSD
/local

/home
/programs
/shared_data

/workdir
/SSD
/local

/home
/programs
/shared_data

/workdir
/SSD
/local

/home
/programs
/shared_data

Network
(10Gb)

Lustre File System
Currently: 1.2 PB

/disk/home

/disk/programs

/disk/shared_data

Backup storage servers

NOTE: Backups not default, need to be configured by users!

Local file systems (fast, few users, temporary on rental servers)

Network file systems (slow, many users, permanent)

Network mounts of local storage between servers

`cbsuX` (hosted server)

On **hosted servers**, the local directory **`/local/storage`** is exported and can be mounted by anyone on any other BioHPC server. On that server, it becomes **network storage**.

/workdir
/SSD
/local
/local/storage

/home
/programs
/shared_data

Anyone can run on **`cbsuY`**:
`/programs/bin/labutils/mount_server cbsuX /storage`

`cbsuY` (any BioHPC machine)

/workdir
/SSD
/local

/fs/cbsuX/storage
/home
/programs
/shared_data

`/local/storage` is
- local on **`cbsuX`**, but...
- ... network-mounted on **`cbsuY`**

Network vs local Storage

Not straightforward to tell which storage is local and which networked just by a name. Remember the setup at BioHPC machines:

- **Networked storage**
  `/home`
  `/shared_data`
  `/programs`
  `/fs/cbsu*/storage`

- **Local storage**
  `/workdir`
  `/local/storage`
  `/SSD`
  `/local_data`

Will look different on other machines or centers – always check description!

Network vs local storage at BioHPC

Feature	Local storage	Network storage	Backup storage
Throughput/speed	high	low if accessed from many nodes simultaneously	Very low
Good as scratch for running programs	yes	no	Absolutely not
Cleaned automatically	Yes – on rental servers only	no	yes
Permanent storage	Yes – on hosted servers only	yes	yes, but old snapshots removed
Direct accessibility (`cd`, `ls`, `cp`, read/write by programs…)	On one machine only, unless network-mounted as **`/fs/cbsu*/storage`**	**`/home/*`** on all BioHPC machines; **`/fs/cbsu*/storage`** where mounted	**`cd`**, **`ls`**, **`cp`** (managed by BioHPC Backup system)
Remote data accessibility (`scp`, `sftp`)	yes	yes	yes (via login nodes)
Cost	In per-hour price (rental) In server cost (hosted)	$98/TB-year First 200GB free (with credit account)	$98/TB-year
Good for backup copies	Yes, on hosted servers	Yes (in **`/home`**)	Yes (designed for it)

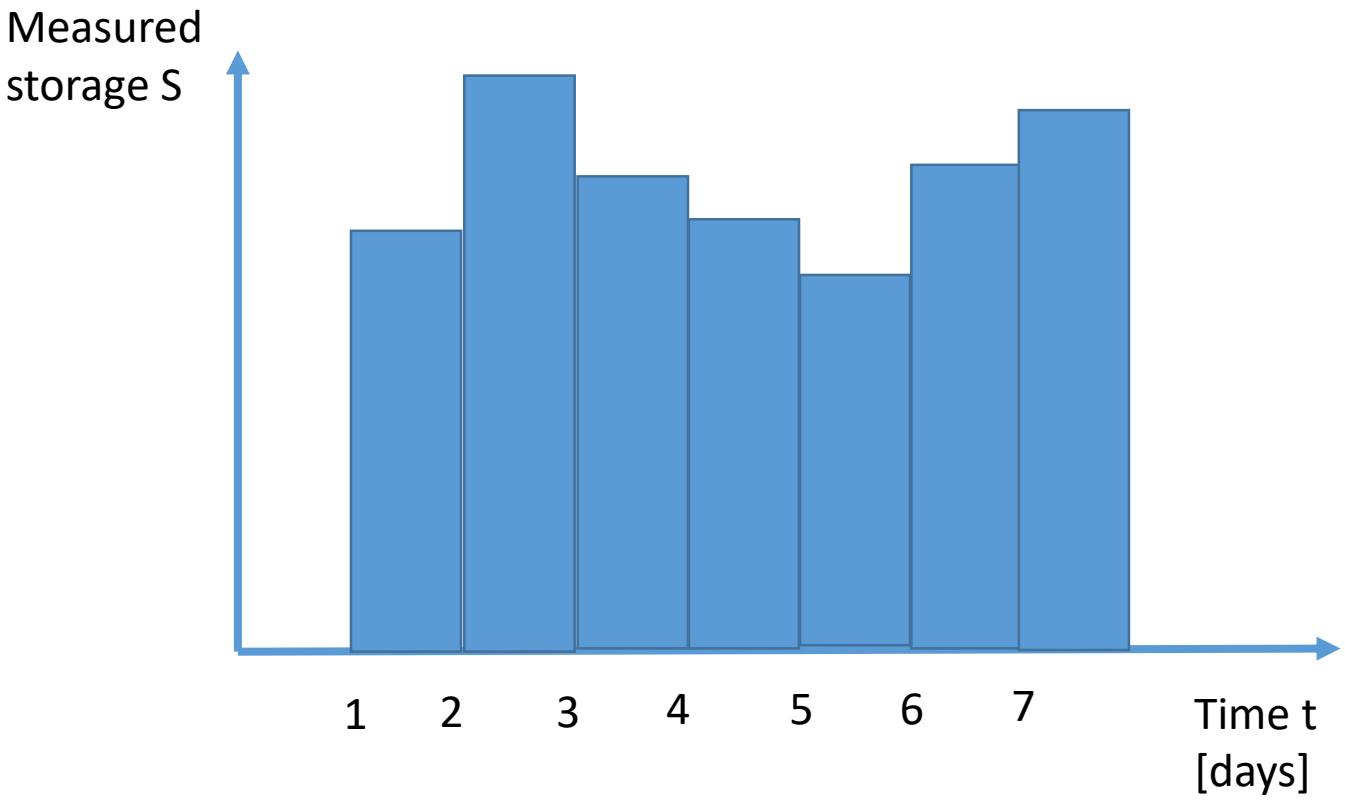
Paying for storage on Lustre (/home) and backup

(biohpc.cornell.edu -> My Storage)

Storage is purchased in **1 TB-year increments** (currently: $98/TB-year)

example: 1 TB-year buys 1 TB of storage for a year, or 2 TB for ½ year, or 0.5 TB for 2 years, etc.

but in reality, disk space occupied changes often, so usage is measured every day and subtracted from the purchased amount:



Usage after N measurements (typically – days)

$$U_N = \sum_{i=1}^{N} S_{i-1} \times \Delta t_i$$

subtracted daily from the originally purchased TB-years, displayed on the website

$\Delta t_i = 1$ day (typically)
$S_0 = 0$ (first day free!)

Are there limits (quota) on BioHPC storage

Local storage

up to physical capacity of local disk array(s)

Network storage (`/home`) – depends on 'involvement' with BioHPC

Users with no storage purchased

20 GB (free) for users without active Credit Account

200 GB (free) for users with/belonging to an active Credit Account

email notification, account lock when limit exceeded, eventually data deleted

Users with storage purchased

no limit (up to Lustre capacity shared among all users)

'warning threshold' – a convenience parameter

email notifications sent when

warning threshold exceeded

number of purchased TB-years is used up (data eventually deleted unless new purchase made)

Data from Network Storage is never deleted automatically without contacting the user(s) involved

Checking storage usage

Network storage on **`/home`** (home directories, group storage directories)

using website: biohpc.cornell.edu - > My Storage

also: purchase storage

using command line, e.g.,

`lfs-du /home/abc123` (will return size of in kB)

Local storage

check total usage (in kB) in a local directory (includes all subdirectories recursively)

`du /workdir/abc123`

check usage (in kB) broken down over subdirectories

`du --max-depth=1 /workdir/abc123`

(need read permissions for the directory sub-tree)

check total, used, and available space on **`/local`** (in kB)

`df -h /local`

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb2       1.6T  777G  762G  51% /local
```

Ways to organize data

Informative directory tree

Group storage space on **/home**

group of users to share certain data – ask brc_bioinformatics@cornell.edu to create

/home/my_group directory (network storage) configured for shared access by the group
option to move group members' HOME directoriess to group storage

Symlinks

pointers (aka shortcuts) to data located in a 'far-away' directory

Example: make a symlink to a folder in group storage in my home directory:

```
cd /home/abc123
ln -s /home/maylabgroup/data labdata
```

Result (when running **ls -al**)

```
lrwxrwxrwx  1 abc123 abc123  2232 Mar 11  2020 labdata -> /home/maylabgroup/data
```

How to move data around BioHPC

Just a few examples. For more information and exercises, refer to Linux workshop

Copy file from network storage to **`/workdir`** (run on a compute machine):

```
cd; cp ./FASTQ/myfile.fastq.gz /workdir/abc123
```

Copy files from a different user's home directory (assume accessible)

```
cd; cp /home/bcd234/shared/*.fastq.gz .
```

Move a directory to a different location (i.e., do **not** keep the original copy)

```
mv /home/abc123/BAMS /local/storage/abc123
```

Make a **copy** of a directory in a different location

```
cp -r /home/abc123/BAMS /local/storage/abc123      OR
rsync -av /home/abc123/BAMS /local/storage/abc123
```

Make a **copy** of a directory on a different machine (called **`cbsuX`**)

```
scp -r /home/abc123/BAMS abc123@cbsuX:/local/storage     OR
rsync -av /home/abc123/BAMS abc123@cbsuX:/local/storage
```

Why use `rsync`?

combines functionality of **`cp`** (local copy) and **`scp`** (remote copy over **`ssh`**)

will (attempt to) preserve owner, group, and permissions

re-startable – will resume from where it was interrupted (just run it again)

configurable with multitude of options, e.g.,

`rsync -av --exclude=*ABC* <source> <destination>`

will skip all objects having ABC in the path

`rsync -avb --backup-dir=/local/oldversion --delete <source> <destination>`

will delete from **`<destination>`** all files absent from **`<source>`**
files deleted from **`<destination>`** and previous versions of those that changed will be placed in **`/local/oldversion`** (on destination server)

user running **`rsync`** must have at least read permissions to the **`<source>`** and write permission to **`<destination>`**

Accessing your BioHPC data from outside

Transfer files from/to an external machine using **sftp**, **scp**; use any SFTP client (e.g., FileZilla, command-line tools) with BioHPC credentials

- login nodes **`cbsulogin.biohpc.cornell.edu`**, **`cbsulogin2.biohpc.cornell.edu`**, **`cbsulogin3.biohpc.cornell.edu`** are accessible from anywhere
  - 2-factor authentication required (unless on campus or on VPN)

- all machines are accessible from Cornell network (including VPN)
  - reservation required to reach machine other than login node

- data available: anything mounted on the server and readable to you
  - **`/home/<your_ID_here>`**
  - **`/fs/cbsu*/storage`** (if mounted)
  - **`/local`**, **`/workdir`** (rental and hosted servers)
  - **`/local/storage`** (hosted servers)

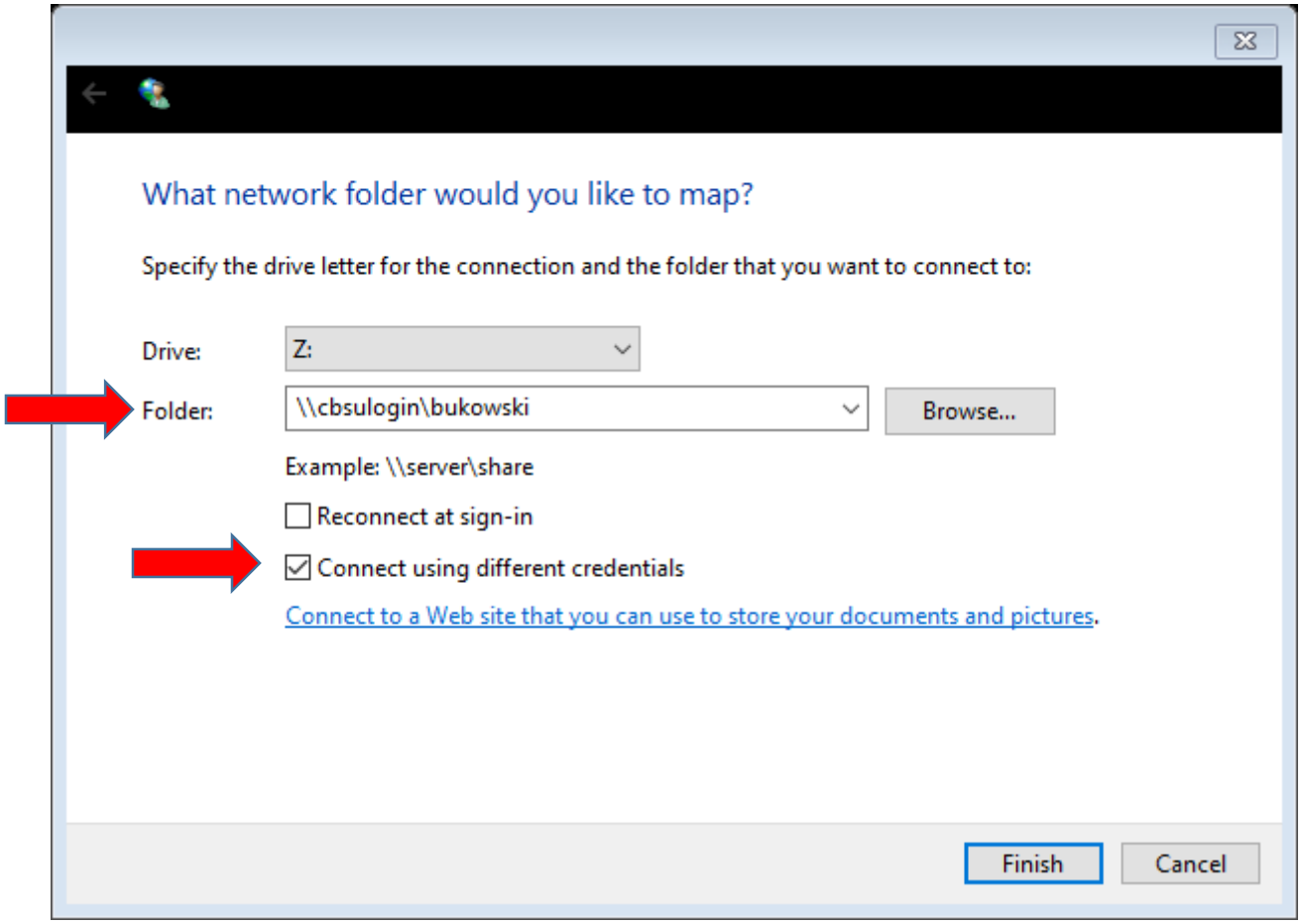
Transfer files between BioHPC and other locations using **Globus**

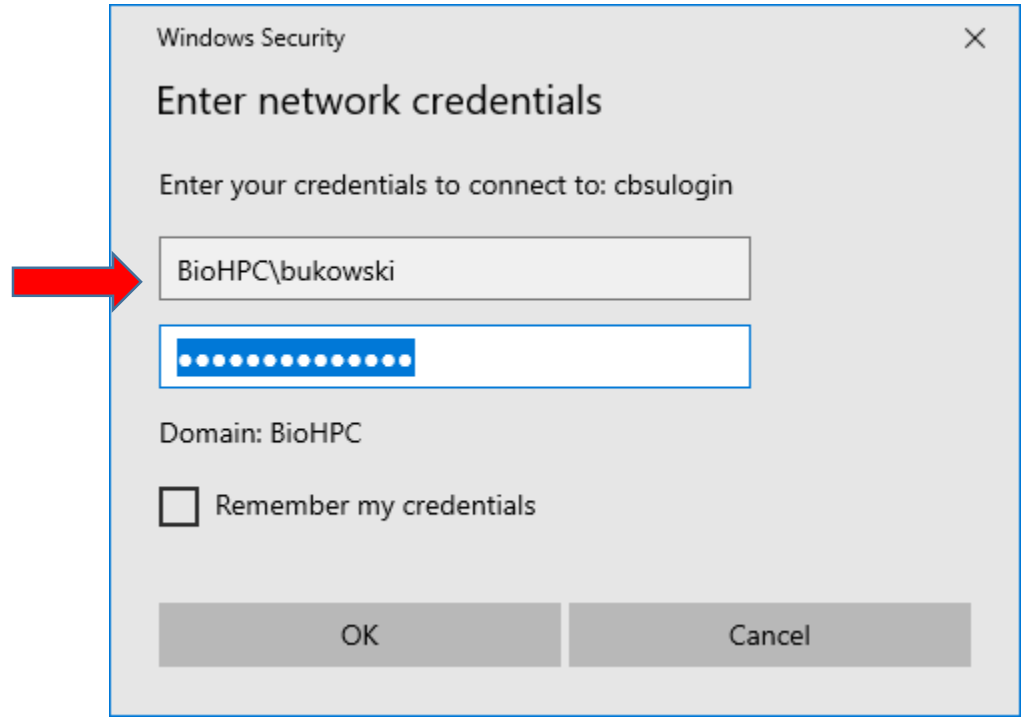
- more on **Globus** later….

Accessing your BioHPC data from outside

Your BioHPC home directory may be accessed directly from Windows or Mac

- ‘Map network drive’ using your BioHPC account as ‘different credentials’:





Backup

Data loss is possible!

- hardware failure
- operator error (accidental deletion or corruption of files)

Be prepared and Back Up!

- All **important data** should exist in at least two independent copies in separate physical locations (servers, rooms, buildings, countries,…)

Remember:

On BioHPC, backups are NOT done by default. It is up to you – the user or a lab – to take action to safeguard your important files!

Backup and Archive

A few things to consider

- What data is important?
- Where to store the copies?
- Keeping copies in sync
    - how often?
    - keep only latest version (mirroring) or also older versions (snapshots) of data?

Immutable data? Consider archiving

- single snapshot, never updated, rarely retrieved
- single copy may be enough (if stored in a super-safe place, e.g., AWS Glacier– intrinsically replicated)
- may need local copy for quick access while processing data

Backup

What is important enough to be backed up? This is up to you, but here are some suggestions/examples:

Important non-mutable	Important mutable	Not important
Raw sequencing data Final results of long analysis (BAM, VCF)	Intermediates from restartable long-running analysis Software under development Publication drafts Derived data from ongoing projects	Scratch/temporary files created by non-restartable software

Where to keep copies (pick 2+)

Option	Best for	Support/ charge by	Auto sync support	Auto snapshots support	Cost ($/TB-year)	Comment
/home	anything	BioHPC	possible	possible	98	First 200 GB free
/local	anything	BioHPC	possible	possible	free	Hosted servers only
BioHPC backup	anything, especially small data changing often	BioHPC	nightly	nightly	98	Anything from /home or /local can be configured by users at https://biohpc.cornell.edu
AWS Glacier Deep Archive	immutable data in large files.	AWS or BioHPC	N/A	N/A	12	User/Lab may use own or BioHPC's AWS account. Archive only, super-safe, but slow and expensive to retrieve. **Internally replicated**.
Public repos	Archiving				free	NCBI, EBI
Cornell Box	Files smaller than 15 GB each	Cornell	-	-	free	Possible to access from BioHPC via **`rclone`**. Rumored to go away in 2023
Shared File System (SFS)	anything	Cornell	native	native	360	Charge per allocated share, regardless of actual use. Possible to mount on BioHPC if formatted as CIFS
GitHub, BitBucket	Source code				depends	Backup, versioning, sharing
Other cloud object storage		vendor	-	-	200-300	AWS S3, Google, Azure (watch for egress charges!), Wasabi (no egress charge, $75/TB-year)
Consumer Cloud Storage		vendor	vendor	vendor	depends	DropBox, OneDrive, Google Drive, …
USB drive on shelf	anything	vendor	vendor	vendor	depends	No intrinsic redundancy (**bitrot** not correctable)

Backup: using BioHPC storage in `/home` and `/local` as copy locations

Example 1

Create a copy of a local directory **`/local/storage/important`** on server **`cbsuXYZ`** on user's **`abc123`** home directory

Log (ssh) in to the machine with data to be copied; optionally launch a **SCREEN** session (or attach an existing one)

- For info on SCREEN – see Question 10 on the FAQ page

Make **`/local/storage`** your 'current directory'

```
cd /local/storage
```

Copy the directory important and recursively all its content to **`/home/abc123`**

```
rsync -av important /home/abc123 >& rsync.log &
```

Check progress

- use **`top`** to see your **`rsync`** process
- **`tail rsync.log`** to see list of recently transferred files

What will be the result?

- directory **`/home/abc123/important`** (a copy of **`/local/storage/important`**)
- record of the operation in file **`/local/storage/rsync.log`** – can be scanned for errors/problems

Backup: using BioHPC storage in `/home` and `/local` as copy locations

Example 2

Create a copy of a local directory **`/local/storage/important`** on server **`cbsuX`** on **`/local/storage`** on another server, **`cbsuY`**

Log (ssh) in to **`cbsuX`**; optionally launch a **SCREEN** session (or attach an existing one)

- For info on SCREEN – see Question 10 on the FAQ page

Make **`/local/storage`** your 'current directory'

```
cd /local/storage
```

Copy the directory important and recursively all its content (you will be asked for password on **`cbsuY`**):

```
rsync -av important cbsuY:/local/storage
```

If you have **passwordless ssh** configured, you can run **`rsync`** in the background: (see question 13 on FAQ page)

```
rsync -av important cbsuY:/local/storage >& rsync.log &
```

What will be the result?

- directory /**`local/storage/important`** on **`cbsuY`** (a copy of **`/local/storage/important`** on **`cbsuX`**)
- record of the operation in file **`/local/storage/rsync.log`** – can be scanned for errors/problems

BioHPC backup service

- Keeps a **periodically updated copy** (mirror) of your selected directories (*backup roots*) on a dedicated server, physically separated from the rest of our infrastructure

  example: home directory or a local directory on a hosted machine

- Keep a number of **previous versions** (snapshots) of each *backup root* directory

  save only files that changed or were deleted – unchanged files are not replicated

- Provide easy access to backed up data

  directories with backups mounted on login nodes (**cbsulogin**, **cbsulogin2,3**)
  backup files can be browsed, searched, and retrieved using standard Linux tools
  ownership and permissions the same as for source directories

Cycle	Source (right before backup run)	On backup server
1	/home/abc/A.txt /home/abc/B.txt /home/abc/C.txt	…../current/home/abc/A.txt …../current/home/abc/B.txt …../current/home/abc/C.txt
2	/home/abc/A.txt /home/abc/B.txt	…../current/home/abc/A.txt …../current/home/abc/B.txt …../bak_Sat_Mar_11_01:46:08_2017/home/abc/B.txt …../bak_Sat_Mar_11_01:46:08_2017/home/abc/C.txt
3	/home/abc/A.txt /home/abc/B.txt	…../current/home/abc/A.txt …../current/home/abc/B.txt …../bak_Sat_Mar_11_01:46:08_2017/home/abc/B.txt …../bak_Sat_Mar_11_01:46:08_2017/home/abc/C.txt …../bak_Sun_Mar_12_01:45:10_2017/home/abc/A.txt

How this works:

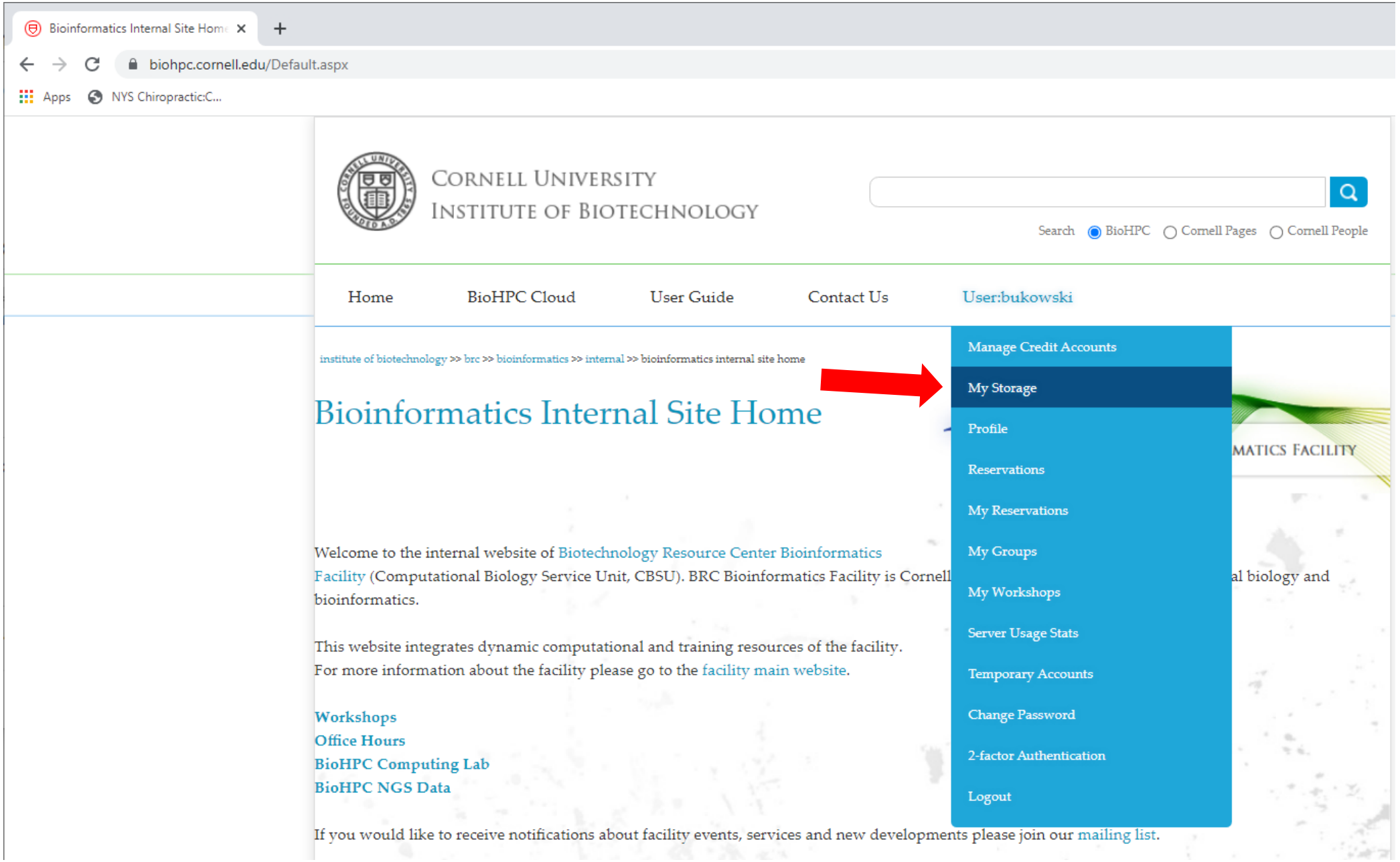
Example:
3 backup cycles for a directory **/home/abc** originally containing 3 files

current: current snapshot
bak_*: changes since last cycle

Different file versions shown in colors

….. path on backup server dedicated to backup root **/home/abc**

Backup is managed through our website https://biohpc.cornell.edu



Backup Credit Account Status

	DATE	Account	Purchased TB-Year	Used TB-Year
Edit Account	1-25-2017	BackupDefaultPool	1.87	55.0917

Backup Storage List

Source Server	Backup Root	Retention	Frequency	MinSave	Current Backup Size(TB)
cbsublfs1	/data1/PanAnd1/RawSeqData/WGS/andropogoneae	10	1	3	2.59
cbsuusda3	/local	10	1	3	4.27
Network Storage	/home/bukowski	10	1	3	0.86
Network Storage	/home/illumina	10	1	3	0.00
Network Storage	/home/imaging_share	10	1	3	23.40

Purchase Backup Credit

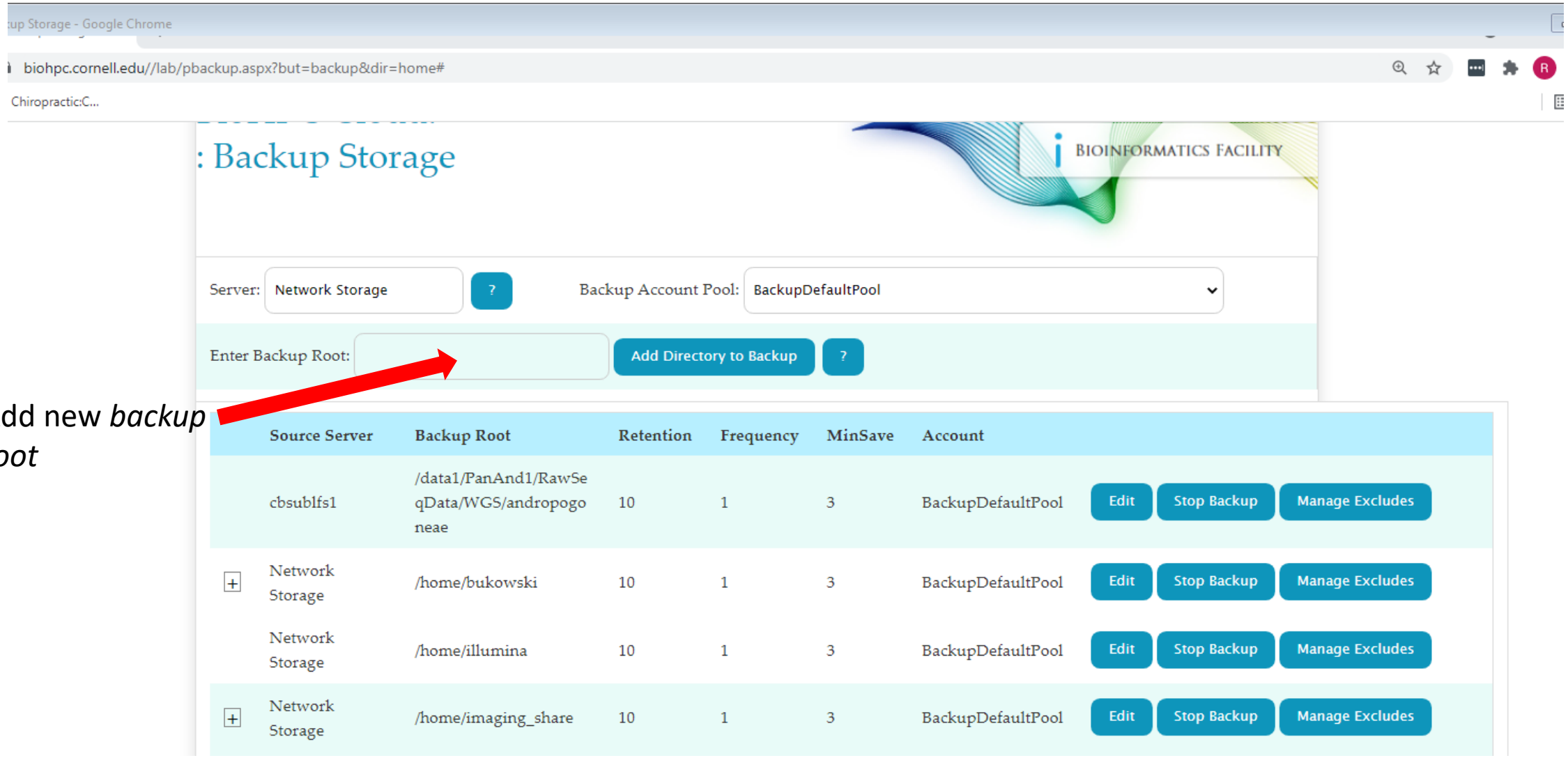
Manage Backup

Website credentials: user: bukowski 'bukowski@cornell.edu' [BioHPC Cloud]
logout

Web Accessibility Help

If not yet done, **purchase backup** storage, put it in Backup Credit Account

Specify **backup root** directories and exclusions



Add new *backup root*

Click to fine-tune exclusions

https://cbsu.tc.cornell.edu//lab/pbackup.aspx?but=backup&dir=home

/home/bukowski/RBackup/logs	Remove Exclude
/home/bukowski/RBackup/logs_from_cbsubgfs1	Remove Exclude
/home/bukowski/RBackup/backup_cron.log	Remove Exclude
/home/bukowski/from_BGISZ	Remove Exclude
/home/bukowski/GATK_tst	Remove Exclude
/home/bukowski/454_2.5.3	Remove Exclude

Back Close List

List	Size	FileType	Exclude
/home/bukowski/454	4096	DIR	☐
/home/bukowski/454_2.5.3	4096	DIR	☑
/home/bukowski/454_2.6	4096	DIR	☐
/home/bukowski/454_2.7	4096	DIR	☐
/home/bukowski/.abrt	4096	DIR	☐
/home/bukowski/AIC-prefs	4096	DIR	☐
/home/bukowski/Amazon_tst	4096	DIR	☐
/home/bukowski/.aspera	4096	DIR	☐
/home/bukowski/aspera_tst	4096	DIR	☐
/home/bukowski/.aws	4096	DIR	☐
/home/bukowski/backups_RB	4096	DIR	☐

Navigate into subfolders

Check to exclude

Accessing backed up files

Directories with backed up data are mounted on the login nodes: **cbsulogin.tc.cornell.edu** and **cbsulogin2.tc.cornell.edu**, currently under **/backups/backup1**. A few examples:

/backups/backup1/bukowski/NetStor/home/bukowski

/backups/backup1/abc123/NetStor/home/abc123_storgrp

/backups/backup1/qisun/NetStor/home/qisun/important_data

/backups/backup1/jarekp/cbsubscb02/local/storage/jarekp

User paying for backup

Server where backup root is located (**NetStor=Network Storage**)

Backup root

Beware: unintentional big backup event triggers to avoid

renaming a large file

equivalent to deleting a file and creating a different one of same size
old-name version will be saved in *bak_**, new version transferred again to *current*

re-organizing a bunch of large files in new subdirectory structures

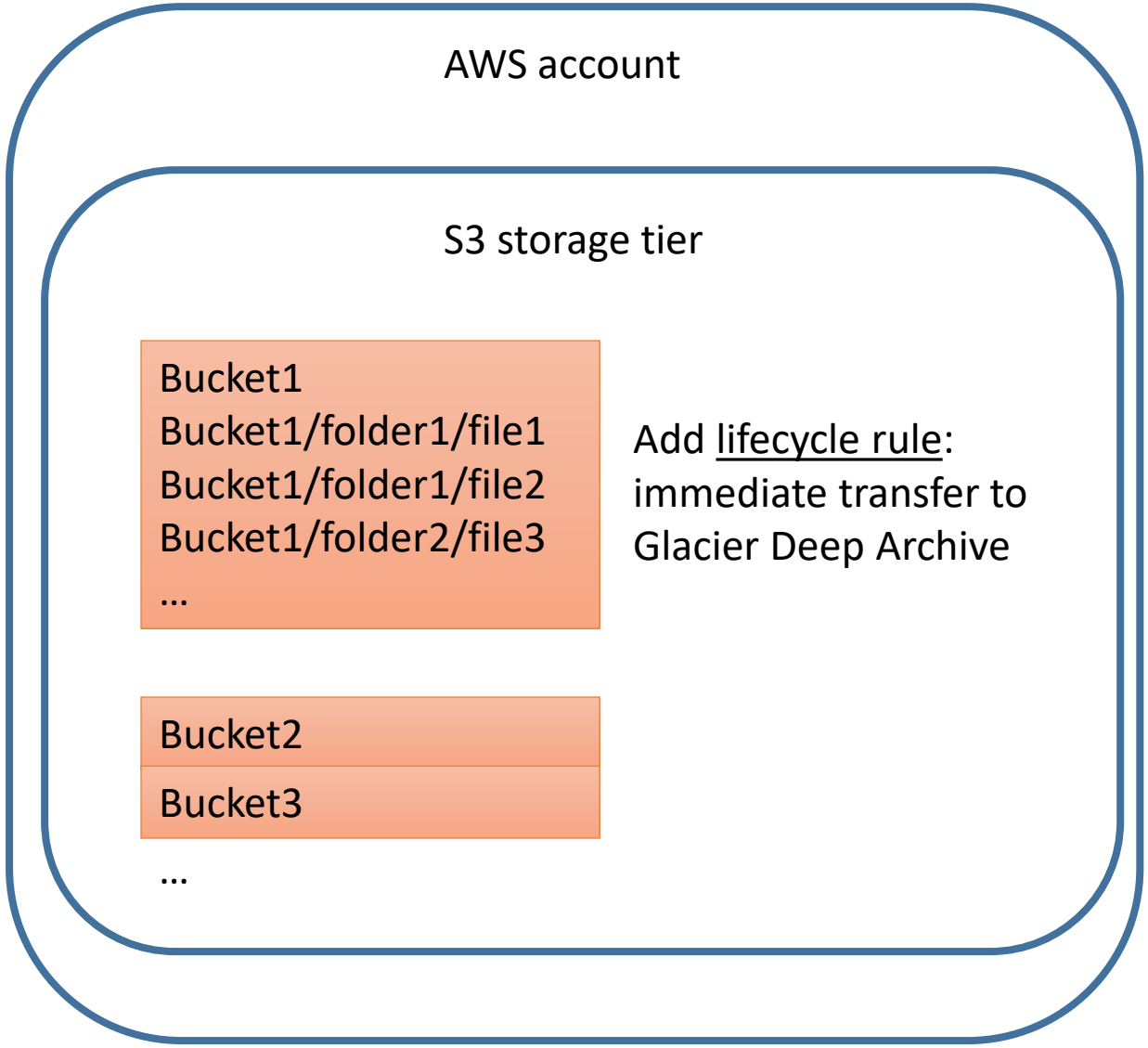
as above

adding one or more large files with no intention to back up, but forgetting to exclude

Consequences

unnecessary network traffic
extra space taken on backup server – user costs incurred

Archiving to AWS Glacier Deep Archive



Obtain/gain access to an **AWS account**

- Option 1: use the AWS account of Bioinformatics Facility (brc_bioinformatics@cornell.edu)
  - Charges assessed by Bioinformatics Facility
- Option 2: get a Cornell-affiliated account by contacting Cornell Cloudification Team
  - Charges assessed by Cornell
- Option 3: get an account directly from AWS with no reference to Cornell (least recommended)
  - Charged directly to AWS

Create a bucket in AWS S3 storage tier with lifecycle rule to transfer to Deep Archive

- if AWS account is through BF, we will create and configure the bucket and give you access

Access the bucket (upload/download/list, …)

- AWS web interface
- **`aws s3`** command-line tool (available at BioHPC)

Archiving to AWS Glacier Deep Archive

PROS

- Cheap ($12/TB-year, pay only for actual usage)
- Reliable (kept triplicate)

CONS

- Slow retrieval
- Retrieval may be expensive ('egress charges')
  - Egress charge $92/TB + retrieval charge $2.5/TB
  - Egress charge waived for Cornell-affiliated AWS accounts
    - If total monthly Cornell egress charge stays below 15% of total Cornell AWS charge – so far always satisfied, but not guaranteed
- Penalty for change or deletion before 180 days

- **Not suitable for large number of small files**
  - Cost of keeping metadata in S3 tier + cost of lifecycle transition may skyrocket
    - 1 TB in 200kB chunks (5 million objects): $24/year (storage) + $250 to get them there (on-time)

  - If data fragmented, need to prepare for Glacier

AWS Glacier Deep Archive is a good option for **immutable** data consisting of large files, with no intention to be retrieved any time soon

Preparing data for AWS Glacier

BioHPC

AWS

Large files (e.g., > 1 GB), preferably compressed

```
/home/abc123/rawdata/*.fastq.gz
/local/storage/BAM/*.bam
```

aws s3 cp
or
AWS Web UI

```
s3://mybucket/rawdata/*.fastq.gz
s3://mybucket/BAM/*.bam
```

Multiple small files

```
/local/storage/tiny/*
```

```
cd /local/storage
tar -czvf tinys.tgz ./tiny
```

/local/storage/tiny.tgz

aws s3 cp
or
AWS Web UI

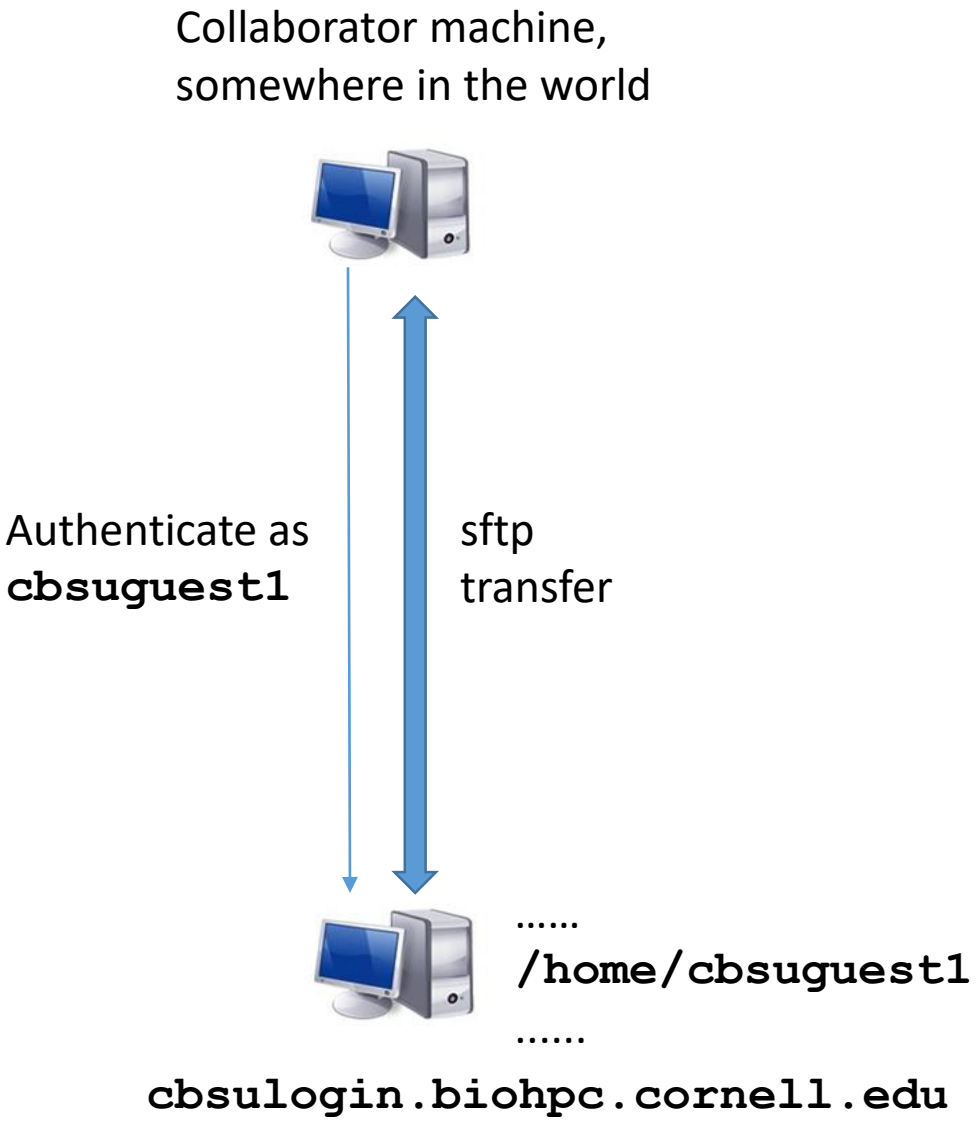
```
s3://mybucket/tinyfiles/tinys.tgz
```

BioHPC can help with this!

Sharing data with outside world

- Sharing via temporary guest accounts
- Sharing via Globus

Temporary guest accounts



You invite the collaborator to temporarily 'take over' a guest account **cbsuguest1**, **cbsuguest2**, etc.

Collaborator gets an e-mail with explanation and link to set password for this account

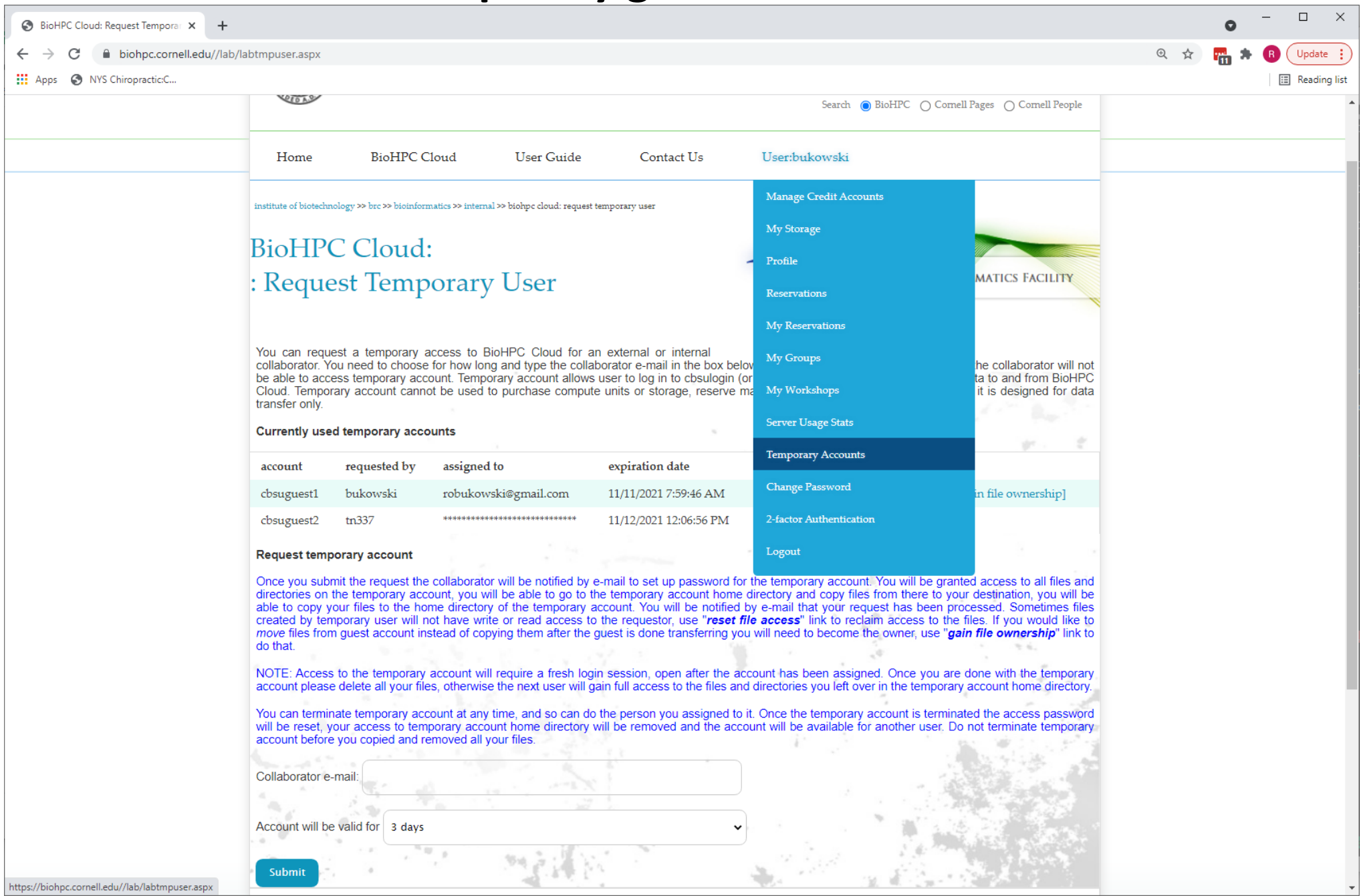
You get full access to the guest account's home directory and can copy data there, or make symlinks to other locations

Collaborator uses sftp client (e.g., FileZilla) to log into **cbsulogin** (or **cbsulogin2**, or **cbsulogin3**) and transfer data in or out

You claim ownership of files deposited by collaborator, copy or move them out to your own storage, then remove them from **/home/cbsuguest1**

You terminate the guest account (or let it expire)

Temporary guest accounts



Sharing via Globus

sftp vs Globus

Log in as **user1**

Host 1
sftp client

/home/user1

Authenticate as
user2

sftp
transfer

/home/user2

Host 2
sftp server, client

/home/user1

Endpoint 1

Authenticate as
user1

GRIDFTP
transfer

globus.org

(may be Endpoint 1)

Log in with your
Globus account
guser0

Authenticate as **user2** (not
needed for access to 'shares')

Endpoint 2
e.g.,
cbsulogin

- **/home/user2**
- 'shares' with Globus users, e.g., with **guser0**

Globus at BioHPC

BioHPC maintains 3 full-featured, licensed endpoints on
`cbsulogin.biohpc.cornell.edu`
`cbsulogin2.biohpc.cornell.edu`
`cbsulogin3.biohpc.cornell.edu`

All endpoints give access to users' home directories (and everything else mounted on login nodes)

Any BioHPC user (including temporary guest users) can access the endpoints

All endpoints allow creating shares

How to share data from BioHPC via Globus

Log in to globus.org (can use your Cornell NetID credentials, or make separate Globus account)

In **Globus File Manager**, in **Collection** box enter **`biohpc#cbsulogin`** (**`cbsulogin2`** or **`cbsulogin3`** will also work)
authenticate with your BioHPC credentials

In **Globus File Manager**, select the directory present on **`cbsulogin`** you want to share
you have to have at least read access to it on **`cbsulogin`**
can be a subdirectory of your **`HOME`** (e.g., **`/home/abc123/myshare`**)
can be a subdirectory of a mount **`/fs/cbsuX/storage`** (where **`cbsuX`** is your hosted server)

Click 'Share' icon, provide share name and other info

Invite other Globus users to the share, defining permissions (read-only or read-write)
invited Globus users will find the share among 'Shared with you'
Globus write permission will work only if the shared directory is writable to you on **`cbsulogin`**

Temporary guest accounts (**`cbsuguest*`**) can access their home directories **`/home/cbsuguest*`** via Globus
can use Globus instead of sftp – good for transferring large data sets

Details (with screenshots): Globus at BioHPC, Sharing BioHPC data via Globus

Sharing BioHPC data among BioHPC users

On Linux, permissions for each file or directory are defined for **three tiers of users**:

- **Owner**: one user who created the file, or has been given ownership by an admin
- **Group**: some group of users; often the **default group** containing only the owner
- **Others**: not owner and not in the file's Group

For each tier of users, permissions are defined by **three attributes** (bits)

- **r** (read)
- **w** (write)
- **x** (for file: execute, for directory: permission to 'cd' into)

```
$ ls -al

drwxr-x---  3 bukowski labgroup  4096 Jun  8 11:33 .
drwxrwxr-x 28 bukowski bukowski  4096 Apr 27  2020 ..
-rw-r-----  1 bukowski bukowski  2232 Mar 11  2020 body.txt
-rwxr--r--  1 bukowski bukowski 15567 Apr 23  2020 CBSU_workshops_export.txt
-rw-r-----  1 irods    panzea     284 Mar 11  2020 download.sh
-rw-r-----  1 bukowski bukowski    58 Jun  8 11:32 emails
-rw-r-----  1 bukowski panzea   17621 Feb 22  2016 file_1.fastq.gz
-rw-r-----  1 bukowski panzea   17200 Feb 22  2016 file_2.fastq.gz
-rw-r-----  1 bukowski labgroup   100 Feb 22  2016 MD5sums
-rwxr-xr-x  1 bukowski labgroup   573 Nov 16  2016 sendmail.pl
drwxrwx---  2 bukowski bukowski  4096 Apr 23  2020 tmp
-rwxr--r--  1 bukowski bukowski  4234 Jul 15  2020 users_July2020.txt
```

directory

attributes owner group

Meaning of permission bits

Bit	Effect on file if set	Effect on dir if set
r	File can be read	Directory content (file and subdir names) can be shown by **ls**
x	File can be executed	One can cd into the directory (**x** required for all subdirs in the path)
w	File can be modified (**x** required for all subdirs in the path) File can be renamed, moved, or removed **only** if **x** is set for all subdirs in the path and **w** is set for parent directory	Files and subdirs can be created, renamed, or removed in the directory [even if there is no **w** on these files themselves (!!)]; **x** also required for all subdirs in the path

NOTE:

To delete a file it is sufficient to have **wx** permission on the parent directory
w permission on the file itself is not needed to delete it

Apart from permissions bits, each file or directory has also three extra bits

Bit	As shown by `ls -al` (example)	Effect on file	Effect on directory
setuid (implies **x**)	`-rwsr-xr-x 1 jarekp cbsuguest1 45583 Feb 12 12:22 some_script.sh`	File will execute as owner (here: jarekp), no matter who runs it	None
setgid (implies **x**)	`drwxr-s--- 4 bukowski cbsuguest1 4096 Feb 12 11:57 my_dir`	File will execute as owning group (here: **cbsuguest1**), no matter who runs it	New files and directories created inside **my_dir** will inherit group (here: **cbsuguest1**); new dirs will have **setgid** set as well
sticky	`-rw-rwxr-t 1 bukowski panzea 172092320 Feb 22 2011 flygenome.fa`	None	File can be deleted or renamed only by the owner, even if **w** on directory allows others to delete/remove files

What happens when an new object is created: group

… in a directory with **`setgid`** bit **not set**

```
drwxr-x---   3 bukowski labgroup  4096 Jun  8  8:33 .
-rw-r--r--   1 bukowski bukowski    58 Jun  8 11:32 newfile
drwxr-xr-x   1 bukowski bukowski  4096 Jun  9 10:12 newdir
```

User who created the object is its **owner**
Group of a new object is the **default group of the owner** (not inherited from directory)

… in a directory with **`setgid`** bit **set**

```
drwxr-s---   3 bukowski labgroup  4096 Jun  8  8:33 .
-rw-r--r--   1 bukowski labgroup    58 Jun  8 11:32 newfile
drwxr-sr-x   1 bukowski labgroup  4096 Jun  9 10:12 newdir
```

User who created the object is its **owner**
Group of a new object is **inherited from directory**
New directory inherits the **`setgid`** bit

What happens when an new object is created: permissions

Permissions for new objects are **independent** of those of the containing directory, i.e., **not inherited**

Permissions for new objects depend un parameter **umask** set individually by each user

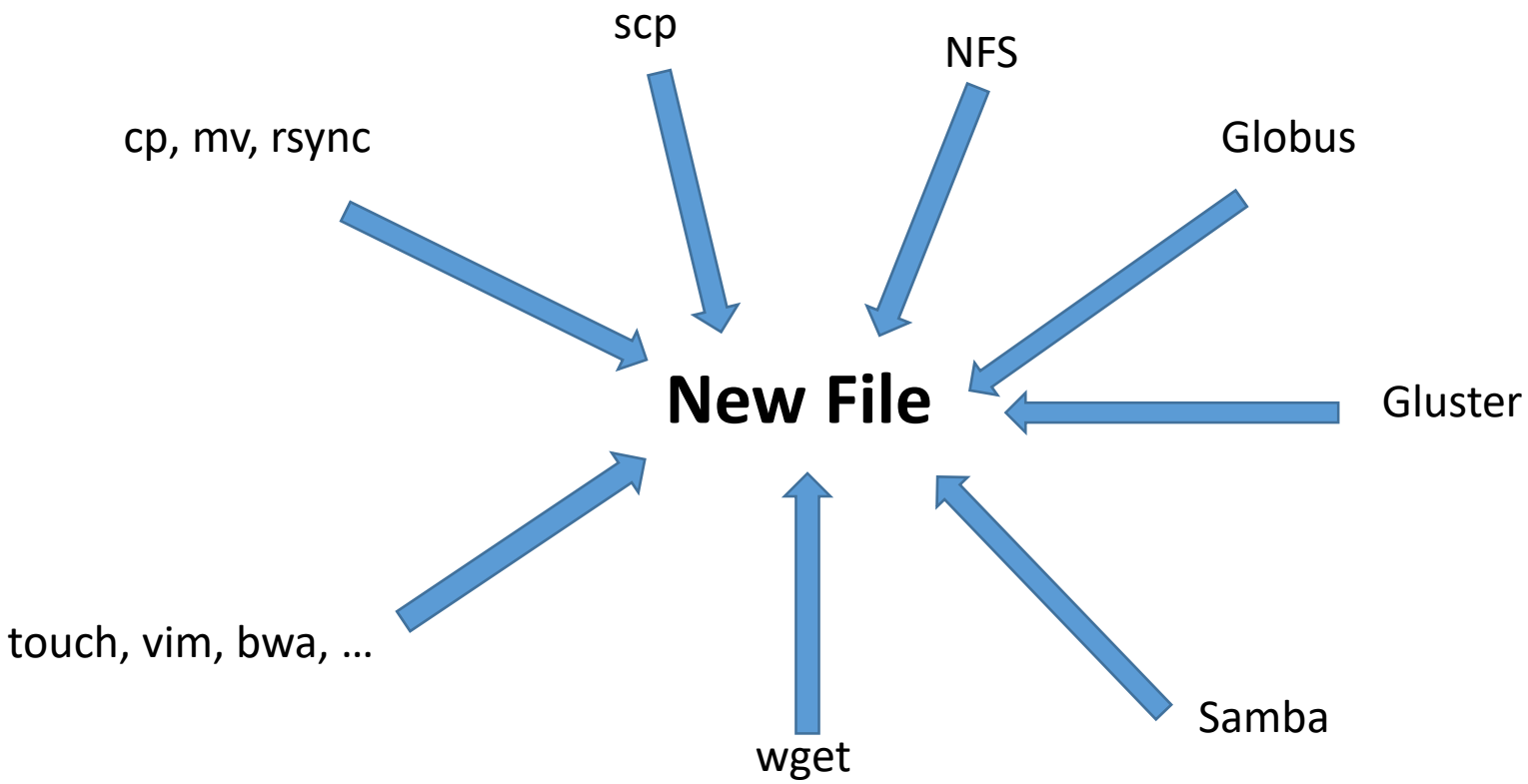
default:
`umask 022` means: **`(rw- r-- r--)`** for new files, **`(rwx r-x r-x)`** for new directories

(some) other possibilities:
`umask 027` means: **`(rw- r-- ---)`** for new files, **`(rwx r-x ---)`** for new directories
`umask 077` means: **`(rw- --- ---)`** for new files, **`(rwx --- ---)`** for new directories
`umask 002` means: **`(rw- rw- r--)`** for new files, **`(rwx rwx r--)`** for new directories
`umask 000` means: **`(rw- rw- rw-)`** for new files, **`(rwx rwx rwx)`** for new directories

Statement **`umask XYZ`** (where **`X,Y,Z=0,2,7,`** etc.) can be put in user's **`$HOME/.bashrc`** file if change in default is needed

Complications

A “new file” can be created by many different tools processes – each with its own “ideas” about ownership and permissions…



Always check/correct permissions set for new files and directories

Basic mechanism for controlling data access (and therefore sharing) on Linux

- Create user groups – done by admin
- Assign users to groups – done by admin
- Assign groups to objects (files and directories)

  tool: Linux command **`chgrp`**
- Assign 'group' and 'others' permission attributes to objects

  tool: Linux command **`chmod`**

As an example, we will present two common scenarios of data sharing using this mechanism

Sharing scenario 1

user **`abc123`** – member of group **`labgroup`** wants to make

folder **`/home/abc123/shared`** readable to the group

folder **`/home/abc123/sharedW`** both readable and writable to the group

folder **`/home/abc123/private`** closed to everyone but the owner

What we need:

```
$ cd /home/abc123; ls -al

drwxr-x---  3 abc123 labgroup     4096 Jun  8 11:33 .
drwxrwxr-x 28   root     root     4096 Apr 27  2020 ..
-rw-r--r--  1 abc123   abc123    15567 Apr 23  2020 CBSU_workshops_export.txt
-rwxr-xr-x  1 abc123   abc123      284 Mar 11  2020 download.sh
drwxrwsr-x  1 abc123 labgroup       58 Jun  8 11:32 sharedW
-rw-r--r--  1 abc123   abc123    17621 Feb 22  2016 file_1.fastq.gz
-rw-r--r--  1 abc123   abc123    17200 Feb 22  2016 file_2.fastq.gz
drwxr-xr-x  1 abc123   abc123      100 Feb 22  2016 shared
-rwxr-xr-x  1 abc123   abc123      573 Nov 16  2016 sendmail.pl
drwx------  2 abc123   abc123     4096 Apr 23  2020 private
```

How to get there:

```
chgrp labgroup .
chmod g=rx,o= .

chmod -R go=rX shared

chgrp -R labgroup sharedW
chmod -R g=rwX sharedW
chmod g+s $(find sharedW -type d)

chmod go= private
```

NOTE:
`umask 002` required to make new objects in **`sharedW`** writable to group

Sharing scenario 2

Make a directory **`/local/storage`** with all its content writable by all members of group **`labgroup`**

What we need:

```
$ cd /local/storage; ls -al

drwxrws---  3   root labgroup  4096 Jun  8 11:33 .
drwxrwxr-x 28   root     root    4096 Apr 27  2020 ..
-rw-rw-r--  1 abc123 labgroup    2232 Mar 11  2020 body.txt
-rwxrw-r--  1 bcd234 labgroup   15567 Apr 23  2020 CBSU_workshops_export.txt
-rwxrwxr-x  1 abc123 labgroup     284 Mar 11  2020 download.sh
drwxrwsr-x  1 abc123 labgroup      58 Jun  8 11:32 data1
-rw-rw-r--  1 bcd234 labgroup   17621 Feb 22  2016 file_1.fastq.gz
-rw-rw-r--  1 bcd234 labgroup   17200 Feb 22  2016 file_2.fastq.gz
drwxrwsr-x  1 cde345 labgroup     100 Feb 22  2016 data2
```

How to get there:

```
chgrp -R labgroup .
chmod g=rws,o= .

chmod g+s $(find . -type d)
chmod -R g=rwX ./*
```

NOTE:
umask 002 for all members of **labgroup** – required for new objects to be group-writable

To discuss your storage, data sharing, or data management needs, contact

brc_bioinformatics@cornell.edu