

POD Resources and Access

- Available PODs
 - Multiple POD group membership
 - POD delegates
- POD access
 - POD compute server shell access
 - POD storage server file transfer access
 - Passwordless Access via SSH/SFTP
 - Samba remote file system access
- POD file systems
 - Home directory
 - Home directory snapshots
 - Home directory quotas
 - Shared Work and Scratch areas
 - Weekly backups
 - Periodic and long-term archiving
- Using POD resources wisely
 - Storage management considerations
 - Manage storage areas by project activity
 - Avoid having too many small files
 - Memory usage considerations
 - Computational considerations
 - Running processes unattended
 - Do not run too many processes
 - Lower priority for large, long-running jobs
 - Multi-processing: cores vs hyperthreads
 - Input/Output considerations
 - Avoid heavy I/O load
 - Transfer large files directly to the storage server
- Other available POD services
 - 3rd party licensed software
 - De-identified protected data
 - Data migration service
 - Temporary data transfer support
 - Instrument backup service

Available PODs

The table below describes the available BRCF PODs, servers and currently available groups. Unless otherwise noted, PODs authenticate using BRCF account credentials initialized by the user in the BRCF account management application (<https://rcrf-account-request.icmb.utexas.edu>).



Anyone with access to a POD may use **any** of the available compute servers, regardless of the server names. For example, both Georgiou and WCAAR users can access **wcarcomp01** and **wcarcomp02**, and both Lambowitz and CCBB users can access **lambcomp01**, **ccbbcomp01** and **ccbbcomp02**.

POD name	Description	BRCF delegates	Compute servers	Storage server	Unix Groups
AMD GPU POD	PUD with GPU resources available for instructional and research use. Note: This POD uses UT EID authentication	Anna Battenhouse	<ul style="list-style-type: none"> • amdgcomp01.ccbb.utexas.edu, amdgcomp02.ccbb.utexas.edu, amdgcomp03.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dual 64-core EPYC 7V13 CPUs ◦ 512 GB RAM ◦ 8 AMD Radeon Instinct MI-100 GPUs w/32GB onboard RAM each 	amdbstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 12 6-TB disks • 72 TB raw, 42 TB usable 	Per course and research project. See <ul style="list-style-type: none"> • The Educational POD

CBRS POD	Shared POD for CBRS core facilities	Anna Battenhouse	<ul style="list-style-type: none"> • cbrscomp01.ccbb.utexas.edu, cbrscomp02.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R640 ◦ dual 26-core/52-thread CPUs ◦ 768 GB RAM ◦ 960 GB SATA SSD for ultra-high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) 	cbrsstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 16-TB disks • 384 TB raw, 220 TB usable 	BCG, CBRS_BIC, CBRS_CryoEM, CBRS_microscopy, CBRS_org, CBRS_proteomics
Chen/Wallingford POD	Shared POD for members of the Jeffrey Chen and John Wallingford labs	<ul style="list-style-type: none"> • Qingxin Song (Chen lab) • Jaime Hibbard (Wallingford lab) 	<ul style="list-style-type: none"> • chencomp01.ccbb.utexas.edu (a.k.a. chencomp02.ccbb.utexas.edu) <ul style="list-style-type: none"> ◦ Dell PowerEdge R410 ◦ dual 4-core/8-thread CPUs ◦ 64 GB RAM 	chenstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 8-TB disks • 192 TB raw, 106 TB usable 	Chen, Wallingford
Dickinson/Cambronne POD	Shared POD for members of the Dan Dickinson and Lulu Cambronne labs	<ul style="list-style-type: none"> • Dan Dickinson • Lulu Cambronne 	<ul style="list-style-type: none"> • djdicomp01.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R410 ◦ dual 4-core/8-thread CPUs ◦ 64 GB RAM 	djdistor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 8-TB disks • 192 TB raw, 106 TB usable 	Dickinson, Cambronne
Educational (EDU)) POD	Dedicated instructional POD Note: This POD uses UT EID authentication	Course instructors. See The Educational POD	<ul style="list-style-type: none"> • edupod.cns.utexas.edu <ul style="list-style-type: none"> ◦ virtual host for pool of 3 physical servers listed below • educcomp01.ccbb.utexas.edu • educcomp02.ccbb.utexas.edu • educcomp04.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R640 ◦ dual 28-core/52-thread CPUs ◦ 1 TB RAM 	educstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 4-TB disks • 96 TB raw, 53 TB usable 	Per course. See The Educational POD

Georgiou WCAAR POD	Shared POD for members of the Georgiou lab and the Waggoner Center for Alcoholism & Addiction Research (WCAAR)	<ul style="list-style-type: none"> Russ Durrett (Georgiou lab) Dayne Mayfield (WCAAR) 	<ul style="list-style-type: none"> wcarcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R430 dual 16-core/32-thread CPUs 256 GB RAM wcarcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R430 dual 18-core/36-thread CPUs 384 GB RAM wcarcomp03.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R640 dual 26-core/52-thread CPUs 1 TB RAM 1.8 TB SATA SSD for ultra-high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) 	georstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 12 8-TB disks + 12 14-TB disks 264 TB raw, 158 TB usable 	Georgiou, WCAAR
GSAF POD	Shared POD for use by GSAF customers. 2TB Work area allocation available for participating groups. Contact Anna Battenhouse, for more information.	<ul style="list-style-type: none"> Anna Battenhouse Dhivya Arasappan 	<ul style="list-style-type: none"> gsafcomp01.ccbb.utexas.edu gsafcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R410 dual 4-core/8-thread CPUs 64 GB RAM gsafcbig01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R720 dual 6-core/12-thread CPUs 192 GB RAM 	gsafstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 24 6-TB disks 144 TB raw, 90 TB usable 	<p>GSAF customer groups: Alper, Atkinson, Baker, Barrick, Bolnick, Bray, Browning, Cannatella, Contreas, Crews, Drew, Dudley, Eberhart, Ellington, GSAFGuest, Hawkes, HoWinson, HyunJunKim, Kirisits, Leahy, Leibold, LiuHw, Lloyd, Manning, Matz, Mueller, Pauli, Press, SSung, ZhangYJ</p> <p>GSAF internal & instructional groups: GSAF, BioComputing2017, CCBB_Workshops_1, FRI-BigDataBio</p>

Hopefog (Ellington) POD	Shared POD for Ellington & Marcotte lab special projects	<ul style="list-style-type: none"> Anna Battenhouse 	<ul style="list-style-type: none"> hfogcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R730xd dual 10-core/20-thread CPUs 250 GB RAM 37 TB local RAID storage, mounted as <code>/raid</code> (not backed up) hfogcomp02.ccbb.utexas.edu, hfogcomp03.ccbb.utexas.edu <ul style="list-style-type: none"> AMD GPU servers 48-core/96-hyperthread EPYC CPU 512 GB RAM 8 AMD Radeon Instinct MI-50 GPUs w/32GB onboard RAM each hfogcomp04.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R750XA dual 24-core/48-thread CPUs 512 GB RAM 2 NVIDIA Ampere A100 GPUs w/80GB onboard RAM each hfogcomp05.ccbb.utexas.edu – available soon! <ul style="list-style-type: none"> GIGABYTE MC62-G40-00 32-core/64-thread AMD Ryzen CPU 512 GB RAM 4 NVIDIA RTX 6000 Ada GPUs, 48G RAM each 	hfogstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 24 6-TB disks 144 TB raw, 90 TB usable 	Ellington, Marcotte, Wilke
Iyer/Kim POD	Shared POD for members of the Vishy Iyer and Jonghwan Kim labs	<ul style="list-style-type: none"> Anna Battenhouse 	<ul style="list-style-type: none"> iyercomp02.ccbb.utexas.edu (aka dragonfly.icmb.utexas.edu) <ul style="list-style-type: none"> Dell PowerEdge R410 dual 4-core/8-thread CPUs 64GB RAM iyercomp03.ccbb.utexas.edu (aka adler3.icmb.utexas.edu) <ul style="list-style-type: none"> Dell PowerEdge R720 dual 6-core/12-thread CPUs 192 GB RAM 	iyerstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 24 6-TB disks 144 TB raw, 90 TB usable 	Iyer, JKim
Kirkpatrick POD	Shared POD for members of Kirkpatrick and Harpak labs	TBD	<ul style="list-style-type: none"> kirkcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R640 dual 26-core/52-thread CPUs 768 GB RAM 1.9 TB SSD for high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) 	kirkstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 12 18-TB disks 216 TB raw, 124 TB usable 	Kirkpatrick, Harpak

Lambowitz /CCBB POD	Shared POD for use by CCBB affiliates and the Alan Lambowitz lab.	<ul style="list-style-type: none"> Hans, Hofmann, Rebecca Young Brim (Hofmann lab & CCBB affiliates) Jun Yao (Lambowitz lab) 	<ul style="list-style-type: none"> lambcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R410 dual 4-core/8-thread CPUs 64 GB RAM ccbbcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R420 dual 4-core CPUs 96 GB RAM ccbbcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R720 dual 6-core/12-thread CPUs 192 GB RAM 	lambstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 18 16-TB disks 288 TB raw, 170 TB usable 	Lambowitz groups: Lambowitz, LambGuest CCBB groups: Cannatella, Hawkes, Hillis, Hofmann, Jansen Instructional groups: FRI-BigDataBio
LiveStrong DT POD	POD for members of Dell Medical School's LiveStrong Diagnostic Therapeutics group. Note: This POD uses UT EID authentication	<ul style="list-style-type: none"> Jeanne Kowalski Song (Stephen) Yi 	<ul style="list-style-type: none"> livecomp01.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R440 dual 14-core/28-thread CPUs 192 GB RAM 480 GB SATA SSD for ultra-high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) livecomp02.ccbb.utexas.edu, livecomp03.ccbb.utexas.edu <ul style="list-style-type: none"> AMD GPU server 48-core/96-hyperthread EPYC CPU 512 GB RAM 8 AMD Radeon Instinct MI-50 GPUs with 32GB onboard RAM each livecomp04.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R640 dual 26-core/52-hyperthread CPUs 768 GB RAM 1.9 TB SSD for high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) 	livestor01.ccbb.utexas.edu <ul style="list-style-type: none"> 24 10-TB disks 240 TB raw, 132 TB usable 	Jeanne Kowalski groups: CancerClinicalGenomics, ColoradoData, MultipleMyeloma Stephen Yi groups: SongYi Lauren Ehrlich groups: Ehrlich_COVID19, Ehrlich Instructional groups: FRI-BigDataBio

Marcotte POD	Single-lab POD for members of the Edward Marcotte lab	<ul style="list-style-type: none"> • Anna Battenhouse 	<ul style="list-style-type: none"> • marccomp01.ccbb.utexas.edu (aka hopper.icmb.utexas.edu) <ul style="list-style-type: none"> ◦ Dell PowerEdge R730 ◦ dual 18-core/36-thread CPUs ◦ 768 GB RAM • marccomp02.ccbb.utexas.edu (aka ada.icmb.utexas.edu) <ul style="list-style-type: none"> ◦ Dell PowerEdge R610 ◦ dual 4-core/8-thread CPUs ◦ 96 GB RAM • marccomp03.ccbb.utexas.edu (aka perutz.ccbb.utexas.edu) <ul style="list-style-type: none"> ◦ Dell PowerEdge R610 ◦ dual 4-core/8-thread CPUs ◦ 96 GB RAM 	marcstor02.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 12-TB disks • 288 TB raw, 160 TB usable 	Marcotte
Ochman/Moran POD	Shared POD for members of the Howard Ochman and Nancy Moran labs	<ul style="list-style-type: none"> • Howard Ochman 	<ul style="list-style-type: none"> • ochmcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R430 ◦ dual 18-core/36-thread CPUs ◦ 384 GB RAM • ochmcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R640 ◦ dual 26-core/52-hyperthread CPUs ◦ 1024 GB RAM ◦ 1.9 TB SSD for high-speed local I/O, mounted as <code>/ssd1</code>, (not backed up) 	ochmstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 24 8-TB disks • 192 TB raw, 106 TB usable 	Ochman, Moran
Rental POD	Shared POD for POD rental customers	<ul style="list-style-type: none"> • Anna Battenhouse (overall) • Daylin Morgan (Brock) 	<ul style="list-style-type: none"> • rentcomp01.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R640 ◦ dual 18-core/36-thread CPUs ◦ 768 GB RAM ◦ 900 GB SATA SSD for ultra-high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) • rentcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> ◦ Dell PowerEdge R640 ◦ dual 18-core/36-thread CPUs ◦ 256 GB RAM ◦ 450 GB SATA SSD for ultra-high-speed local I/O, mounted as <code>/ssd1</code> (not backed up) 	rentstor01.ccbb.utexas.edu <ul style="list-style-type: none"> • 12 12-TB disks • 144 TB raw, 90 TB usable 	Brock, Calder, Champagne, Curley, Fleet, Gaydosh (AddHealth, FragileFamilies, VUSNAPS), Gray, Gross, Hillis, Raccach, Seidlits, Sullivan, YiLu, Zamudio

Wilke POD	For use by members of the Claus Wilke lab and the AG3C collaboration	<ul style="list-style-type: none"> Aaron Feller Alexis Hill 	<ul style="list-style-type: none"> wilkcomp01.ccbb.utexas.edu wilkcomp02.ccbb.utexas.edu <ul style="list-style-type: none"> Dell PowerEdge R930 quad 14-core /28-thread CPUs 1 TB RAM wilkcomp03.ccbb.utexas.edu <ul style="list-style-type: none"> GIGABYTE MC62-G40 48-core AMD Ryzen 5975 CPU 500 G system RAM 4 NVIDIA RTX 6000 Ada GPUs, 48G RAM each 2 TB SSD for fast local I/O, mounted as <code>/ssd1</code> (not backed up) 	wilkstor01.ccbb.utexas.edu <ul style="list-style-type: none"> 18 16-TB disks 288 TB raw, 170 TB usable 	Wilke
---------------------	----------------------------------------------------------------------	-------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------	--------------

Multiple POD group membership

Depending on your affiliations, you may have access to more than one POD. For example, you may have active accounts on both the Lambowitz/CCBB POD and the GSAF POD.

You may also belong to more than one group on a given POD. For example, you may belong to both the **Hofmann** and **GSAF** groups on the GSAF POD. These POD groups control which shared Work and Scratch areas you can access.

To see what groups you are a member of, use the [groups](#) shell command from any POD compute server. For example:

```
$ groups
Hofmann GSAF
```

The **first** group in this list is your current group, which determines the group ownership for files you create. To change your current group so that new files are marked with a different group, use the [newgrp](#) shell command. For example:

```
$ groups
Hofmann GSAF

$ newgrp - GSAF
$ groups
GSAF Hofmann
```



Your primary POD group

Note that your **primary** (default) POD group is the group that is active (first in the [groups](#) list) when you first log on to a POD server.

If you would like to change your default POD group, please [Contact Us](#).

You can also change the group assigned to existing files/directories using the [chgrp](#) command. For example, to make sure all files and directories under a particular directory are associated with a specific group, you would execute this command:

```
chgrp -R Hofmann <path_to_directory>
```

POD delegates

POD delegates act as local liaison to the BRCF for member organizations. Their responsibilities include:

- Communicate and help enforce BRCF policies among their colleagues
- Approve requests for user accounts
- Recommend user and group quotas
- Implement and monitor sub-directory organization in shared Work and Scratch areas
- Evaluate requests for additional software installations and communicate such to BRCF

- may also perform test installations to evaluate new software functionality
- May have administrative rights on compute servers to assist their POD users with common issues (e.g. permissions)

POD access

By directive of UT's Information Security Office (ISO), starting April 1, 2019, access to BRCF POD resources (including [ssh](#)) will be blocked from outside the UT campus network. See [UT VPN service setup](#) for how to set up the UT Virtual Private Network (VPN) service to enable off-campus access or access from the Dell Medical School.

The table below lists the available POD resources and how they can be accessed. In addition, this [remote_computing_software_download_instruction_s.pdf](#) PDF provides detailed information about how to configure the UT VPN service, set up Duo 2-factor authentication, and installing software for remote SSH access in Windows.

Resource	Description	Network availability	For details
SSH	Remote access to the bash shell's command line, and remote file transfer commands such as scp and rsync .	<ul style="list-style-type: none"> • Standard ssh command unrestricted from the UT campus network (excluding Dell Medical School) • Off-campus ssh access: <ul style="list-style-type: none"> ◦ UT VPN service active, or ◦ Public key installed in <code>~/.ssh/authorized_keys</code> • Notes: <ul style="list-style-type: none"> ◦ Direct storage server access for file transfers are only accessible from the UT campus network or with the UT VPN service active. 	<ul style="list-style-type: none"> • POD server shell access • POD file transfer access • Passwordless remote SSH access
Samba	Allows mounting of shared POD storage as a remote file system that can be browsed from your Windows or Mac desktop/laptop computer	<ul style="list-style-type: none"> • Unrestricted from the UT campus network (excluding Dell Medical School) • Off-campus access requires the UT VPN service to be active 	<ul style="list-style-type: none"> • Samba remote file system access
HTTPS	Access to web-based R Studio server and JupyterHub applications	<ul style="list-style-type: none"> • Unrestricted for BRCF-managed accounts <ul style="list-style-type: none"> ◦ For PODs using EID authentication (e.g. Livestrong), an active UT EID is required 	<ul style="list-style-type: none"> • Web-based RStudio and Python • About R and R Studio Server

POD compute server shell access

Compute servers can be accessed via [ssh](#) using either their formal BRCF name or their alias. For Mac and Linux users, [ssh](#) is available from any Terminal window. For Windows users, any SSH client program can be used, such as PuTTY (<http://www.putty.org/>).

```
ssh <brcf_account>@<compute_server>.ccbb.utexas.edu

# for example
ssh abattenh@cbrscomp01.ccbb.utexas.edu
```

[ssh](#) access is available from UT campus network addresses, with public key encryption (see below) or by using the [UT VPN service](#).



Networks at Dell Medical School are not part of the UT campus network, so require or public keys or the [UT VPN service](#).

POD storage server file transfer access

BRCF storage servers do not offer interactive shell ([ssh](#)) access. However, they provide direct file transfer capability via [scp](#) or [rsync](#). Using the storage server as a file transfer target is useful when you have many files and/or large files, as it provides direct access to the shared storage.

See this FAQ for usage tips: [Transferring data to/from PODs](#)

Direct storage server file transfers are available from UT campus network addresses or using the [UT VPN service](#).

Passwordless Access via SSH/SFTP

You can set up password-less access to the pod nodes via [ssh](#) from a specific, trusted machine (your office machine, laptop, another POD, TACC, etc).

To set up password-less [ssh](#) access from any Linux-like environment (e.g. Mac Terminal, cygwin on Windows, Windows 10 Linux subsystem), follow the steps below. If you are using Windows [PuTTY](#) see this documentation: <https://linux-audit.com/using-ssh-keys-instead-of-passwords/>.

1. On the machine and account you want to **ssh/sftp from** (e.g. your laptop), generate a SSH key pair if you don't already have one:

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
cd ~/.ssh
if [ ! -f id_rsa ]; then ssh-keygen -b 4096 -t rsa; fi
```

Use the default answers for **ssh-keygen**, and do **not** specify a password. This creates a public/private key pair in your local **~/.ssh** directory (**id_rsa.pub** and **id_rsa**, respectively).

2. Install your public key on the server you want to login **to** (e.g. any one of your POD compute nodes)
 - a. If you are off campus and do not have access to the UT VPN service, [Contact Us](#) via email, so we can install it for you.
 - include your BRCF account name and attach your public key (**~/.ssh/id_rsa.pub**).
 - b. If you are on campus or have access to the [UT VPN service](#), you can use the **ssh-copy-id** command:

```
ssh-copy-id user@hostname
```

- If you are setting this up from off campus, you need to have the [UT VPN service](#) active for this command to work remotely.
- If you are prompted to accept the SSH Host Key for the node you are connecting to, type "yes" to do so.
- You will be asked for your password for the user@hostname, so enter it when prompted.

3. Login to the machine to make sure it is working properly:

```
ssh user@hostname
# requirement to use non-standard port 222 removed June 2022
```

- If you are prompted for a password, then something went wrong with the setup.
 - Most likely, it is file permissions on your local or remote home directory
 - You must **not** have group or world write access to your home directory or your **~/.ssh** directory).
 - If you have multiple SSH keys (RSA, DSA, etc) on the machine you are connecting **from** then it could also be using the wrong key to connect.

Samba remote file system access

The Samba remote file system protocol allows you to mount POD storage from your desktop or laptop as if it were a local file system. Samba access is available from UT network addresses or using the [UT VPN service](#). In addition, this [remote_computing_software_download_instructions.pdf](#) PDF provides detailed information about how to configure the UT VPN service, set up Duo 2-factor authentication, and installing software for remote SSH access in Windows.



Some networks at Dell Medical School are not part of the UT campus network, so require use of the [UT VPN service](#).

Samba access to your Home directory and to shared Work areas is available on most PODs for most POD groups.

- The Samba server name is the POD's storage server name as listed above.
- The Samba share name for your **Home** directory is always **users**.
- The Samba share name for shared **Work** areas is the group name.
- Be sure to provide your BRCF account credentials to authenticate (by default your laptop or desktop account is used)

For Mac users, Samba resource access syntax is of the form **smb://<server_name>/<share_name>**. For example:

- **smb://gsafstor01.ccbb.utexas.edu/users** – Samba share for an individual Home directory on the GSAF POD.
- **smb://gsafstor01.ccbb.utexas.edu/GSAF** – Samba share for an the **GSAF** group's shared Work directory on the GSAF POD.

To connect to your Group's Work area as a network volume on a Mac:

- Go To Finder (click on the desktop or on the clown icon in the taskbar)
- In Finder:
 - Select Go menu item, then Connect to Server
 - Enter URL: **smb://gsafstor01.ccbb.utexas.edu/GSAF**
 - Connect
 - You'll see a dialog asking "You are attempting to connect to the server "xxx"
 - Select Connect
 - You'll see an "Enter your name and password" dialog
 - In "Enter your name and password" dialog, enter your BRCF account name and password
 - Select Connect
- To return to this folder later, go back to Finder
 - In the sidebar, scroll past Favorites to the Locations section
 - Select your server name from there

For Windows users, Samba resource access syntax is of the form `\\<server_name>\<share_name>`. For example:

- `\\gsafstor01.ccbb.utexas.edu\users` – Samba share for an individual Home directory on the GSAF POD.
- `\\gsafstor01.ccbb.utexas.edu\GSAF` – Samba share for an the **GSAF** group's shared Work directory on the GSAF POD.

To connect to your Group's Work area as a network drive in Windows:

- Bring up Windows Explorer (**Windows key-E**)
- On Windows 10:
 - You'll see "**Computer**" in the menu
 - You'll see "**Map network drive**" in the sub-menu
 - Select "**This PC**" icon
 - Select "**Computer**" menu item
- On Windows 7:
 - You'll see "**Map network drive**" in the menu
 - Select "**Computer**" icon
- Click "**Map network drive**".
- In the "**Map Network Drive**" dialog
 - Select a drive letter
 - In the "**Folder**" text box, enter your Group area URL
 - e.g. for the **Sullivan** group on the GSAF pod:
`\\gsafstor01.ccbb.utexas.edu\Sullivan`
 - Check the "**Connect using different credentials**" checkbox
 - Enter your BRCF account name and password
 - If your computer is on the UT Austin Active Directory Domain you need to add "." before your BRCF account name
 - e.g. `.mybrcfaccount`
 - Click "**Finish**". This will bring up the "**Enter Network Password**" dialog
- In the "**Enter Network Password**" dialog
 - Select "**Use another account**"
 - Enter your BRCF user name and password
 - Check the "**Remember my credentials**" checkbox if desired
 - Click "**OK**"
 - A new Windows Explorer will appear with your Work area in focus

POD file systems

All of the POD compute servers have access to their own shared storage, where each user has an individual Home directory and shared Work and Scratch areas.

Home directory

Your Home directory on a POD is located under `/stor/home`. Home directories are meant for storing small files. **All home directories have a 100 GB quota.**

By default you are only allowed access to your own Home directory, although you may be able to view Home directory contents for other members of your group depending on the group's permissions policy.

Home directories are backed up weekly, and have snapshots enabled.

Home directory snapshots

Read-only snapshots are periodically taken of your home directory contents. Like Windows backups or Macintosh Time Machine, these backups only consume disk space for files that change (are updated or deleted), in which case the previous file state is "saved" in a snapshot.

Snapshots are stored in a `.zfs/snapshot` directory under your home directory. To see a list of the snapshots you currently have:

```
ls ~/.zfs/snapshot
```

To recover a changed or deleted file, first identify the snapshot it is in, then just copy the file from that snapshot directory to its desired location.

Home directory quotas

Your 100 GB Home directory includes snapshot data. These snapshot backups only consume disk space for files that change (are updated or deleted), in which case the previous file state is "saved" in the snapshot. Snapshots are taken frequently, so their data persists for several months even if the associated Home directory file has been deleted.

The main consequence of this snapshot behavior is that they can cause your 100 GB Home directory quota to be exceeded, even after non-snapshot files have been removed.

While you can view and copy files in your `~/.zfs/snapshot` snapshot directories, you cannot write to them or delete them. Please contact us at rctf-support@utexas.edu to remove your snapshots if you exceed your Home directory quota.

Note that some Home directory sub-directories can become quite large, such as `~/.local/share/rstudio` where session files can be very big and cause your Home directory quota to be exceeded. A solution is to copy directories like this to your Scratch area and symbolic link to them. For example:

```
mkdir -p /stor/scratch/BCG/abattenh/home_extra
rsync -avrP ~/.local/share/rstudio/ /stor/scratch/BCG/abattenh/home_extra/local_rstudio/
rm -rf ~/.local/share/rstudio
ln -sf /stor/scratch/BCG/abattenh/home_extra/local_rstudio/ ~/.local/share/rstudio
```

Another case involves using [R Studio Server](#) or [JupyterHub Server](#) on a POD. Both these web-based applications by default use Home directories as the default working directory. This is not an issue as long as files created there are relatively small, but data directories for larger projects should be located in Work or Scratch. Navigation to these areas can be simplified using symbolic links as shown below.

Similarly, if large files are first copied to Home (e.g. when transferred from TACC), then moved to Work or Scratch, they may still take Home directory snapshot space if a snapshot is taken before they are moved (and snapshots are taken frequently). To avoid this issue, always transfer files directly to Work or Scratch. You can create a symbolic link to these areas in your Home directory to help with this. For example:

```
# change to your home directory where the symlinks will be created
cd
ln -s -f /stor/work/BCG bcg_work
ln -s -f /stor/scratch/BCG bcg_scratch

# Then, use the symbolic link when copying data from TACC
rsync -avrW $SCRATCH/analysis/ abattenh@cbrsstor01.ccb.utexas.edu:~/bcg_scratch/analysis/
```

Shared Work and Scratch areas

Shared Work and Scratch areas are available for each POD group under `/stor/work/<GroupName>` and `/stor/scratch/<GroupName>` (for example, `/stor/work/Hofmann`, `/stor/scratch/Hofmann`). These areas are accessible only by members of the named group. Users can find out which group or groups they belong to by typing the `groups` command on the command line.

These Work and Scratch areas are designed for storage of shared project artifacts, so they have no predefined structure (i.e. user directories are *not* automatically created). Group members may create any directory structure that is meaningful to the group's work.

Shared Work areas are backed up weekly. Scratch areas are not backed up. Both Work and Scratch areas may have quotas, depending on the POD (e.g. on the Rental or GSAF pod); such quotas are generally in the multi-terabyte range.

Because it has a large quota and is regularly backed up and archived, your group's Work area is where large research artifacts that need to be preserved should be located.

Scratch, on the other hand, can be used for artifacts that are transient or can easily be re-created (such as downloads from public databases).

See [Manage storage areas by project activity](#) for important guidelines for Work and Scratch area contents.

Weekly backups

All Home and Work directories are backed up weekly to a separate backup storage server (spinning disk). Backups take place sometime between Friday and Monday mornings and are currently **not** incremental backups.

Note that any directory in any file system tree named `tmp`, `temp`, or `backups` **is not backed up**. Directories with these names are intended for temporary files, especially large numbers of small temporary files. See ["Cannot create tempfile" error](#) and [Avoid having too many small files](#).

Periodic and long-term archiving

Data on the backup server are periodically archived to TACC's **Ranch** tape archive roughly once a year. Current archives are as of:

- 2022-01
- 2020-07

In addition, to avoid re-archiving the same directories multiple times, we maintain a "long term archives (LTA)" directory that contains data from projects that are no longer active. Such project data may have been transferred to a group's Scratch area to avoid consuming backup space, or may have been removed from POD storage entirely after archiving to avoid consuming storage server space.

Please [Contact Us](#) if you need something retrieved from tape archives.

Using POD resources wisely

Remember that PODs are shared resources, and it is important to be aware of how your work can affect others trying to use POD resources. Here are some tips for using POD resources wisely.

Storage management considerations

Manage storage areas by project activity

Shared POD storage servers are high capacity (~50 to ~250 TB), but space is not infinite! The same goes for backup storage, since the BRCF must have capacity to back up all POD Home and Work areas. The following guidelines will help you and your colleagues stay within storage limits.

There are several types of data activity that determine where the data should reside:

- Data that is **active**, such as project directories where new files are added and ongoing analysis is taking place.
 - This data belongs in your Work area where it is regularly backed up.
- Data that is **no longer active**, or is active but read-only) but **needs to be accessible** for reference, and **needs to be preserved**.
 - E.g. projects that are complete but that you refer to from time to time.
 - This data belongs in your Scratch area so that it does not consume backup space.
 - Please contact us at rctf-support@utexas.edu to request that a long-term archive of the data be made to tape.
 - We can also efficiently move the data from Work to Scratch for you since we can access the storage server directly.
- Data that is **no longer active** and **does not need to be referenced**, but **needs to be preserved**.
 - This data can be removed entirely so that it does not consume either storage server or backup server space.
- **External/public data or downloaded software** that **needs to be accessible** but **does not need to be backed up or preserved**.
 - This data always belongs in Scratch since it can be re-downloaded if necessary.
- Data that is **no longer active**, **does not need to be referenced**, and **does not need to be preserved**.
 - You can delete this data yourself, or contact us to remove the data for you (we can do this efficiently since we can access the storage server directly).

This table summarizes these guidelines.

#	active?	external?	needs to be accessible?	needs to be preserved?	examples	process/actions
1	yes	no	yes	TBD	<ul style="list-style-type: none">• current project & analysis directories that are read and written	<ul style="list-style-type: none">• store in regularly backed-up Work area
2	no	no	yes	yes	<ul style="list-style-type: none">• no-longer-active projects that still need to be referenced• read-only data such as FASTQ or other instrument-generated files	<ul style="list-style-type: none">• store in Scratch area• contact rctf-support@utexas.edu to create a tape archive copy and to move the directories from Work to Scratch for you
3	no	no	no	yes	<ul style="list-style-type: none">• no-onger-active projects that do not need to be readily accessible	<ul style="list-style-type: none">• contact rctf-support@utexas.edu to create a tape archive copy for you, then remove the data (either from Work or Scratch)
4	yes	yes	yes	no	<ul style="list-style-type: none">• data and annotations from public databases• downloaded software	<ul style="list-style-type: none">• always store in Scratch area, since this is external data that can be re-downloaded if necessary
5	no	yes or no	no	no	<ul style="list-style-type: none">• abandoned projects• external data or software that is no longer deleted	<ul style="list-style-type: none">• delete the directories/files yourself, or contact rctf-support@utexas.edu to delete it for you

Avoid having too many small files

While the ZFS file system we use is quite robust, we can experience issues in the weekly backup and periodic archiving process when there are too many small files in a directory tree.

What is too many? Ten million or more.

If the files are small, they don't take up much storage space. But the fact that there are *so many* causes the backup or archiving to run for a *really* long time. For weekly backups, this can mean that the previous week's backup is not done by the time the next one starts. For archiving, it means it can take weeks on end to archive a single directory that has many millions of small files.

Backing up gets even worse when a directory with many files is just moved or renamed. In this case the files need to be deleted from the old location and added to the new one – and both of these operations can be extremely long-running.

To see how many files (termed "inodes" in Unix) there are under a directory tree, use the **df -i** command. For example:

```
df -i /stor/work/MyGroup/my_dir
```

The results might look something like this:

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
stor/work/MyGroup	103335902213	28864562	103307037651	1%	/stor/work/MyGroup

The **IUsed** column (here 28864562) is the number of inodes (files plus directories) in the directory tree listed under **Filesystem** (here `/stor/work/MyGroup`). Note that the reported **Filesystem** may be different from the one you queried, depending on the structure of the ZFS file systems.

There are a several work-arounds for this issue.

1) Move the files to a temporary directory.

The backup process excludes **any sub-directory anywhere** in the file system directory tree named **tmp**, **temp**, or **backups**. So if there are files you don't care about, just rename the directory to, for example, **tmp**. There will be a one-time deletion of the directory under its previous name, but that would be it.

2) Move the directories to a Scratch area.

Scratch areas are not backed up, so will not cause an issue. The directory can be accessed from your Work area via a symbolic link. Please [Contact Us](#) if you would like us to help move large directories of yours to Scratch (we can do it more efficiently with our direct access to the storage server).

3) Zip or Tar the directory

If these are important files you need to have backed up, **zipping** or **taring** the directory is the way to go. This converts a directory and all its contents into a single, larger file that can be backed up or archived efficiently. Please [Contact Us](#) if you would like us to help with this, since with our direct access to the storage server we can perform **zip** and **tar** operations much more efficiently than you can from a compute server.

If your analysis pipeline creates many small files as a matter of course, you should consider modifying the processing to create small files in a **tmp** directory then **zipping** or **taring** the as a final step.

Memory usage considerations

Using too much RAM can quickly make a compute server unusable. When a system's main random access memory (RAM) is filled and additional memory requests are made, "pages" of main memory will be written out to "swap" space on disk, then read back in when again needed. Since disk I/O is on the order of 1,000 times slower than RAM access, swapping can slow a system down considerably.

And in a pathological (but unfortunately not uncommon) pattern, a program (or programs) that need more memory than available can cause "**thrashing**" where swapping in and out of RAM is happening continuously. This will bring a computer to its knees, making it virtually impossible to do anything on it (slow logins, or logins timing out; any simple command just "hanging" for a long time or never returning). We monitor system usage, and will intervene when we see this happen, by terminating the offending process(es) if possible, or by rebooting the compute server if not.

You can avoid causing a problem like this by following this advice:

Tips:

- Know the memory configuration of the compute server you're using
 - **free -g** will show you total RAM and swap in Gigabytes
- Before starting a memory intensive job, check the system's current memory status
 - **free -g** also shows used and available for both main memory and swap
- Know the memory requirements of your program.
 - Monitor its memory usage while it is running using top (see <https://www.booleanworld.com/guide-linux-top-command/>)
 - This is particularly important if you plan to run multiple instances of a program, since it will guide you in knowing how many such instances you should run.
- Run memory intensive processes when system load is otherwise light (e.g. overnight)

Computational considerations

Running processes unattended

While POD compute servers do not have a batch system, you can still run multiple tasks simultaneously in several different ways.

For example, you can use terminal multiplexer tools like **screen** or **tmux** to create virtual terminal sessions that won't go away when you log off. Then, inside a **screen** or **tmux** session you can create multiple sub-shells where you can run different commands.

You can also use the command line utility **nohup** to start processes in the background, again allowing you to log off and still have the process running.

Here are some links on how to use these tools:

- **nohup** - <http://linux.101hacks.com/unix/nohup-command/>
- **screen** - <https://kb.iu.edu/d/acuy>
- **tmux** -
 - <http://tmuxcheatsheet.com/>
 - <https://gist.github.com/MohamedAlaa/2961058>

Do not run too many processes

Having described how to run multiple processes, it is important that you **do not run too many processes at a time**, because you are just using one compute server, and you're not the only one using the machine!

How many is "too many"? That really depends on what kind of job it is, what compute/input-output mix it has, and how much RAM it needs. **As a general rule, don't run more simultaneous jobs on a POD compute server than you would run on a single TACC compute node.**

Before running multiple jobs, you should check RAM usage (**free -g** will show usage in GB) and see what is already running using the **top** program (press the **1** key to see per-hyperthread load), or using the **who** command, or with a command like this:

```
ps -ef | grep -v root | grep -v bash | grep -v sshd | grep -v screen | grep -v tmux | grep -v 'www-data'
```

Here is a good article on all the aspects of the **top** command: <https://www.booleanworld.com/guide-linux-top-command/>

Finally, be sure to lower the priority of your processes using **renice** as described below (e.g. **renice -n 15 -u `whoami`**).

Lower priority for large, long-running jobs

If you have one or more jobs that uses multiple threads, or does significant I/O, its execution can affect system responsiveness for other users.

To help avoid this, please use the **renice** tool to manipulate the priority of your tasks (a priority of 15 is a good choice). It's easy to do, and here's a quick tutorial: http://www.thegeekstuff.com/2013/08/nice-renice-command-examples/?utm_source=tuicool

For example, before you start any tasks, you can set the default priority to nice 15 as shown here. Anything you start from then on (from this shell) should inherit the nice 15 value.

```
renice +15 $$
```

Once you have tasks running, their priority can be changed for all of them by specifying your user name:

```
renice +15 -u `whoami`
```

or for a particular process id (PID):

```
renice +15 -p <some PID number>
```

Multi-processing: cores vs hyperthreads

Many programs offer an option to divide their work among multiple processes, which can reduce the total clock time the program will run. The option may refer to "processes", "cores" or "threads", but actually target the available computing units on a server. Examples include: **samtools sort --threads** option; **bowtie2 -p/--threads** option; in **R**, `library(doParallel); registerDoParallel(cores = NN)`.

One thing to keep in mind here is the difference between **cores** and **hyperthreads**. Cores are **physical** computing units, while hyperthreads are **virtual** computing units -- kernel objects that "split" each core into two hyperthreads so that the single compute unit can be used by two processes.

The **AvailablePODs** table describes the compute servers that are associated with each BRCF pod, along with their available cores and (hyper)threads. (Note that most servers are dual-CPU, meaning that total core count is double the per-CPU core count, so a dual 4-core CPU machine would have 8 cores.) You can also see the hyperthread and core counts on any server via:

```
cat /proc/cpuinfo | grep -c 'core id'          # actually the number of hyperthreads!
cat /proc/cpuinfo | grep 'siblings' | head -1  # the real number of physical cores
```

(Yes, the fact that 'core id' gives hyperthreads and 'siblings' the number of cores is confusing. But what do you expect -- this is Unix 😊)

Since hyperthreads **look like** available computing units ("CPUs" in OS displays), parallel processing options that detect "cores" usually really detect hyperthreads. Why does this matter?

The bottom line:

- **virtual Hyperthreads** are useful if the work a process is doing periodically "yields", typically to perform input/output operations, since waiting for I/O allows the core to be used by other work. Many NGS tools fall into this category since they read/write sequencing files.
- **physical Cores** are best used when a program's work is compute-bound. When processing is compute bound -- as is typical of matrix-intensive machine learning algorithms -- hyperthreads actually **degrade** performance, because two compute-bound hyperthreads are competing for the same physical core, and there is OS-level overhead involved in process switching between the two.

So before you select a process/core/thread count for your program, consider whether it will perform significant I/O. If so, you can specify a higher count. If it is compute bound (e.g. machine learning), be sure to specify a count low enough to leave free hyperthreads for others to use.

Note that this issue with machine learning (ML) workflows being incredibly compute bound is the main reason ML processing is best run on GPU-enabled servers. While none of our current PODs have GPUs, GPU-enabled servers are available at TACC. Additionally, Austin's Advanced Micro Devices, who are trying to compete with NVIDIA in the GPU market, will soon be offering a "GPU cloud" that will be available to UT researchers. We're working with them on this initiative and will provide access information when it is available.

Input/Output considerations

Avoid heavy I/O load

Please be aware of the potential effects of the input/output (I/O) operations in your workflows.

Many common bioinformatics workflows are largely I/O bound; in other words, they do enough input/output that it is essentially the gating factor in execution time. This is in contrast to simulation or modeling type applications, which are essentially compute bound.

It is underappreciated that I/O is much more difficult to parallelize than compute. To add more compute power, one can generally just increase the number of processors, their speed, and optimize their CPU-to-memory architecture, which greatly affects compute-bound tasks.

I/O, on the other hand, is harder to parallelize. Large compute clusters such as TACC expose large single file system namespaces to users (e.g. Work, Scratch), but these are implemented using multiple redundant storage systems managed by a sophisticated parallel file system (Lustre, at TACC) to appear as one. Even so, file system outages at TACC caused by heavy I/O are not uncommon.

In the POD architecture, all compute servers share a common storage server, whose file system is accessed over a high-bandwidth local network (NFS over 10 Gbit ethernet). This means that **heavy I/O to shared storage initiated from any compute server can negatively affect users on all compute servers**.

For example, as few as three simultaneous invocations of [gzip](#) or [samtools sort](#) on large files can degrade system responsiveness for other users. If you notice that doing an [ls](#) or command completion on the command line seems to be taking forever, this can be a sign of an excessive I/O load (although very high compute loads can occasionally cause similar issues).

To gauge your program's I/O usage:

1. Run it on smaller datasets first
2. Check I/O effects by exercising tab-completion from the command line (see below)
 - a. tab completion is directly impacted by I/O load, so if it slows there's too much I/O going on

```
ls /st                # Typing this + Tab expands to /stor
ls /stor/sy           # Typing this + Tab expands to /stor/system
ls /stor/system/o     # Typing this + Tab expands to /stor/system/opt
ls /stor/system/opt/sam # Typing this + Tab expands to /stor/system/opt/samtools (not uniquely)

# Typing this + Tab twice will list many possible completions:
ls /stor/system/opt/samtools/bam
```

Transfer large files directly to the storage server

BRCF storage servers are just Linux servers, but ones you access from compute servers over a high-speed internal network. While they are not available for interactive shell ([ssh](#)) access; they provide direct file transfer capability via [scp](#) or [rsync](#).

Using the storage server as a file transfer target is useful when you have many files and/or large files, as it provides direct access to the shared storage. Going through a compute server is also possible, but involves an extra step in the path – from the compute-server to its network-attached storage-server.

The solution is to target your POD's storage server directly using [scp](#) or [rsync](#). When you do this, you are going directly to where the data is physically located, so you avoid extra network hops and do not burden heavily-used compute servers.



Note that direct storage server file transfer access is only available from UT network addresses, from TACC, or using the UT VPN service.

Please see this FAQ for more information: [I'm having trouble transferring files to/from TACC](#).

Other available POD services

3rd party licensed software

We can, upon request, install 3rd party licensed software on your POD as long as an appropriate license is provided to us, and the software is compatible with our standard Ubuntu operating system. Please contact us at rcf-support@utexas.edu if you have such a request.

De-identified protected data

While our PODs are not HIPAA-compliant, we do support de-identified data that is considered sensitive/protected by the data provider, and we can provide appropriate information and protocols describing how we protect such data in data protection plan paperwork required by external data providers. These measures include:

- Protecting sensitive data directories using Unix group permissions, where group membership is controlled by the PI
- Disabling backups of sensitive data directories
- Disallowing Samba remote file system access to sensitive data directories
- Providing a group-specific temporary directory

Please contact us at rctf-support@utexas.edu if you would like to host protected data on your pod.

Data migration service

Upon request we can assist with transferring large datasets to your POD storage from other locations, such as external hard drives, servers at other institutions, or TACC. Please contact us at rctf-support@utexas.edu if you have such a request.

Temporary data transfer support

Since researchers often need to exchange large datasets with external collaborators, upon request, we can set up an area on your POD where an external collaborator can **temporarily** read/write data. Please contact us at rctf-support@utexas.edu if you have such a request, providing contact information for the external collaborator.

Instrument backup service

Many instruments have a Windows computer that controls the instrument and stores its data. Upon request we can set up automated and/or manual backups of data on Windows instrument-associated machines, using the Samba remote file system protocol to access a group-specific storage area on your POD. Please contact us at rctf-support@utexas.edu if you have such a request.