

Intro to Data Transfer with ARCC

Introduction: This workshop will introduce the core concepts behind options for data storage and transfers to and from ARCC resources. After going through this workshop, participants should have a better understanding of the tools available to transfer data from their local desktop to ARCC resources as well as different types of storage systems. Covered in this workshop are the risks involved with data storage and transfer, best practices to mitigate associated risks, options available for transferring data between ARCC resources, how to determine which options work best, and how to use different options. Examples will be given for both Graphical User Interfaces (GUI) as well as Command Line Interfaces (CLI). These topics are modular and do not need to be followed in order, so please feel free to jump to the topic you are most interested in.

Goals:

- Using web-based data transfer applications
- Using a desktop client to transfer data
- Using a command line interface
- Interactive connections to data storage

This workshop assumes participants have an understanding of command line Linux

Sections:

1. [Using the Open OnDemand Web Interface to transfer data](#)
 2. [Basic Data Transfer with Globus](#)
 3. [Other Globus Options](#)
 4. [Data Transfer with Desktop Clients](#)
 5. [Command Line Data Transfer](#)
 6. [Mapping/Mounting with SMB](#)
-

Data Transfer with Open OnDemand

With Open OnDemand being an official ARCC supported service, we would be remiss to not cover how Data Transfer works within the application. The Open OnDemand web portal includes a file browsing option that enables file management capabilities for users to upload, download, copy/move, view, rename, and delete files on ARCC HPC and Storage resources. However, there are some nuances that users should be aware of before committing to using Open OnDemand as their only data transfer tool. Please follow along with the examples below to learn about the data transfer abilities of Open OnDemand.

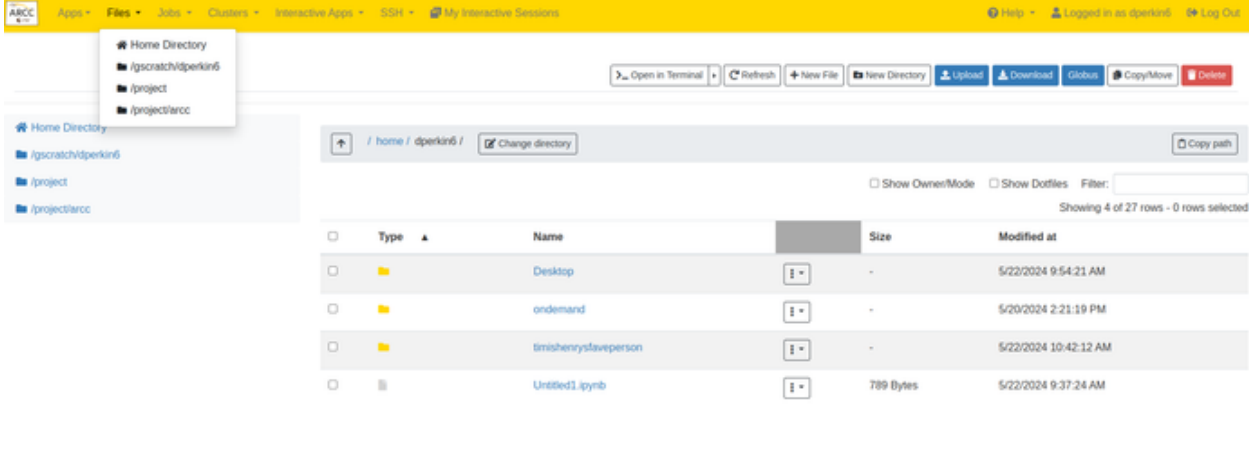
Before we begin, please make sure to have logged into Open OnDemand for MedicineBow just as you have in a prior tutorial, by navigating your web browser to <https://medicinebow.arcc.uwyo.edu/>.

-
- [The File Browser App](#)
 - [Exploring the App](#)
 - [Uploading a File](#)
 - [Open OnDemand Restrictions](#)
 - [Next Steps](#)
-

The File Browser App

Easily found on the top bar of the Open OnDemand Interface, the Files App can take you to one of your three main cluster directories of home, gscratch, and project. From here you can perform several file operations:

- upload
- download
- Create new files/directories
- copy/move
- rename
- Delete



The screenshot shows the Open OnDemand Files App interface. The top navigation bar includes 'Apps', 'Files', 'Jobs', 'Clusters', 'Interactive Apps', 'SSH', and 'My Interactive Sessions'. A dropdown menu is open under 'Files', showing 'Home Directory', '/gscratch/dperkin6', '/project', and '/project/arc'. The main content area shows the current directory path as '/ home / dperkin6 /'. Below this is a table of files and directories:

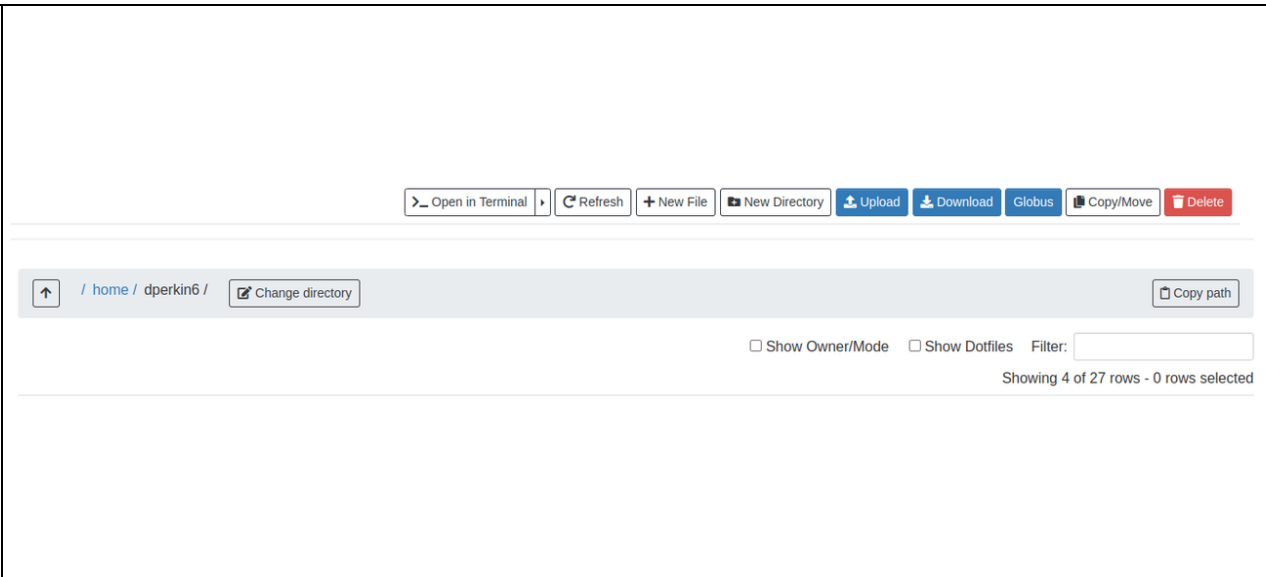
Type	Name	Size	Modified at
Directory	Desktop	-	5/22/2024 9:54:21 AM
Directory	ondemand	-	5/20/2024 2:21:19 PM
Directory	trishenystaveperson	-	5/22/2024 10:42:12 AM
File	Untitled1.ipynb	789 Bytes	5/22/2024 9:37:24 AM

Below the table, it says 'Showing 4 of 27 rows - 0 rows selected'. At the bottom of the screenshot, there is a note: 'It's important to know that file permissions can't be changed in Open On Demand'.

Exploring the App

As you can see in the top image, there are buttons, check boxes, and a search bar available to perform many functions previously mentioned and some that haven't been.

It is important to know that Open OnDemand has their own terminology for hidden files called Dotfiles. If for some reason you have transferred some configuration files but aren't seeing the, you can check this box to see them listed.

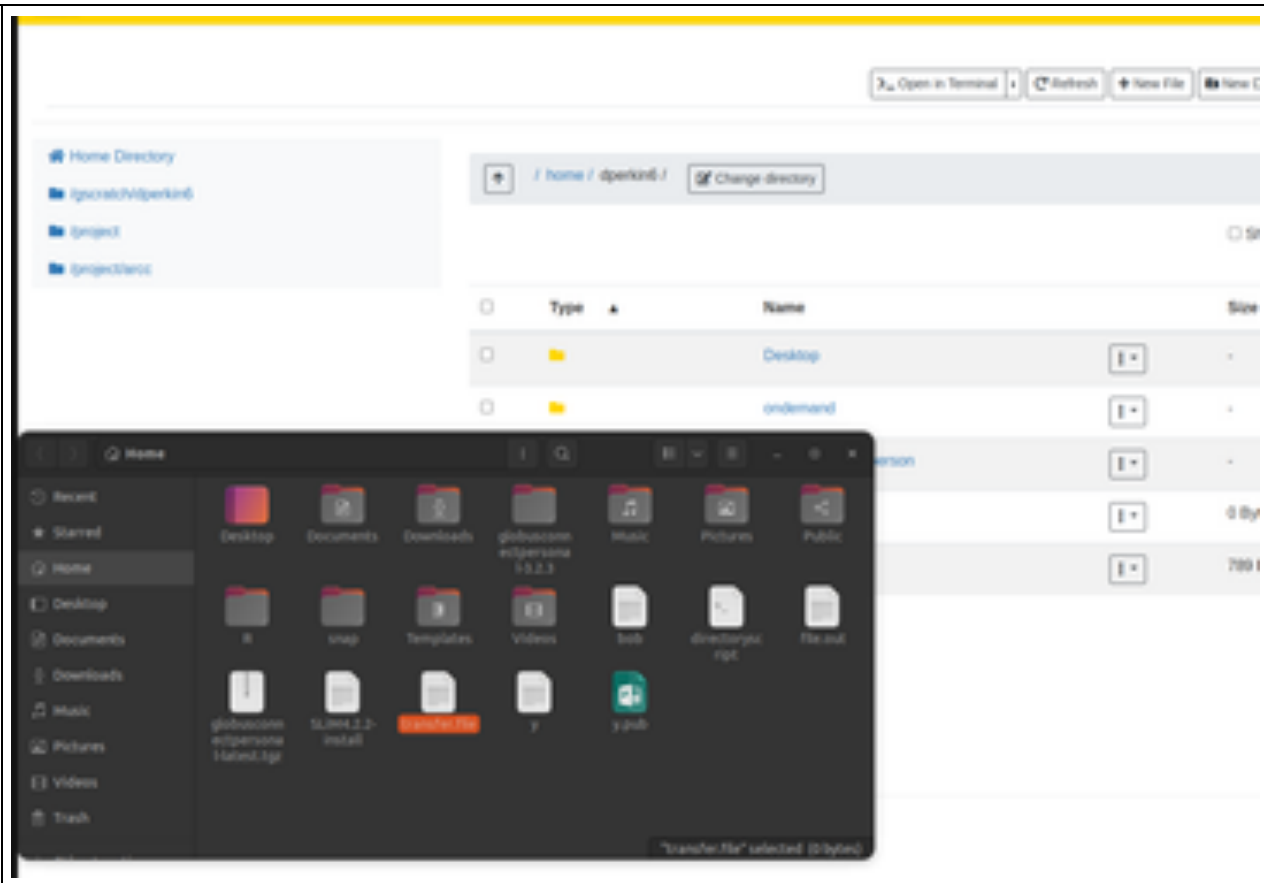


Uploading a File

There are two ways to upload data.

- You can drag and drop from your file browser into the Files App
- Or you can Click on the 'Upload' Button

This works for both entire directories as well as individual files.



Open OnDemand Restrictions

While Open OnDemand makes transferring data as easy as other tools you may be used to like Google Drive or Dropbox, it is a tool that was designed to do much more than just transfer data and since it is using the https protocol. ARCC only recommends transferring small files such as code, README files or anything under 5 gigabytes.

Do not transfer large files with Open OnDemand. Please see the next section on Globus for transferring large amounts of data.

If you are unsure if Open OnDemand will work for your data transfer needs, please see the other tools that ARCC supports or contact us at arcchelp@uwyo.edu

Basic Data Transfer with Globus

Globus is ARCC's recommended method of transferring data onto our resources. Globus provides a secure, unified interface to your research data. Use Globus to 'fire and forget' high-performance data transfers between systems within and across organizations. With Globus, users can do more beyond just transfer data Globus users can move, share, & transfer, sync, and find data via a single interface regardless of where the data actually "lives". Whether your files are housed on a supercomputer, lab cluster, tape archive, public cloud or your laptop, you can manage this data from anywhere, using your existing identities, accessible through your web browser.

- [Globus Key Concepts](#)
 - [Globus Web Interface](#)
 - [Logging into Globus](#)
 - [Account Linking](#)
 - [Globus File Manager](#)
 - [ARCC's Collections](#)
 - [Basic Globus Transfer](#)
 - [Enable Globus Personal Endpoint](#)
 - [Next Steps](#)
-

Globus Key Concepts

It's important to be aware of a key concepts that only apply to Globus transfers. This will clear up much of the confusion for the content discussed below.

- **Collection** - A collection is a named location containing data you can access with Globus. Collections can be hosted on many different kinds of systems, including campus storage, HPC clusters, laptops, Amazon S3 buckets, Google Drive, and scientific instruments. When you use Globus, you don't need to know a physical location or details about storage. You only need a collection name. A collection allows authorized Globus users to browse and transfer files. Collections can also be used for sharing data with others and for enabling discovery by other Globus users.
- **Endpoint** - An endpoint is a server that hosts collections. If you want to be able to access, share, transfer, or manage data using Globus, the first step is to create an endpoint on the system where the data is (or will be) stored. An endpoint can be a laptop, a personal desktop system, a laboratory server, a campus data storage service, a cloud service, or an HPC cluster. It's easy to set up your own Globus endpoint on a laptop or other personal system using [Globus Connect Personal](#). Administrators of shared services (like campus storage servers) can set up multi-user endpoints using Globus Connect Server. You can use endpoints set up by others as long as you're authorized by the endpoint administrator or by a collection manager.
- **Fire-And-Forget Data Transfer** - After you request a file transfer, Globus takes over and does the work on your behalf. You can navigate away from the File Manager, close the browser window, and even logout. Globus will optimize the transfer for performance, monitor the transfer for completion and correctness, and recover from network errors and collection downtime. When a problem is encountered part-way through the transfer, Globus resumes from the point of failure and does not retransmit all of the data specified in the original request. Globus can handle extremely large data transfers, even those that don't complete within the authentication expiration period of a collection (which is controlled by the collection administrator). If your credentials expire before the transfer completes, Globus will notify you to re-authenticate on the collection, after which Globus will continue the transfer from where it was paused.

Globus Web Interface

The first step in transferring data with Globus is to navigate to the Globus web site at <https://www.globus.org/>. From there click on the LOG IN icon.



Logging into Globus

To begin using Globus with ARCC systems, use the option for logging in with your existing organizational login and search for the “University of Wyoming”.

- Caveat: If the account does not have a UWYO associated e-mail, you will not be able to set up globus with that account.

Log in to use Globus Web App

Use your existing organizational login

e.g., university, national lab, facility, project

[privacy policy.](#)

Continue

Account Linking

After logging in with your UWyo credentials, you will be prompted with a message about linking accounts. This is something that should be looked into if you have used Globus at another institution of if you have a separate Globus ID.

Welcome – You've Successfully Logged In

This is the first time you are accessing Globus with your **University of Wyoming** login.

If you have previously used Globus with another login you can link it to your **University of Wyoming** login. When linked, both logins will be able to access the same Globus account personal history.

[Continue](#)

[Link to an existing account](#)

[Why should I link accounts?](#)

Globus File Manager

The Globus File Manager is the primary interface to transferring files. On this screen is where we begin looking for collections that we want to transfer from or to.

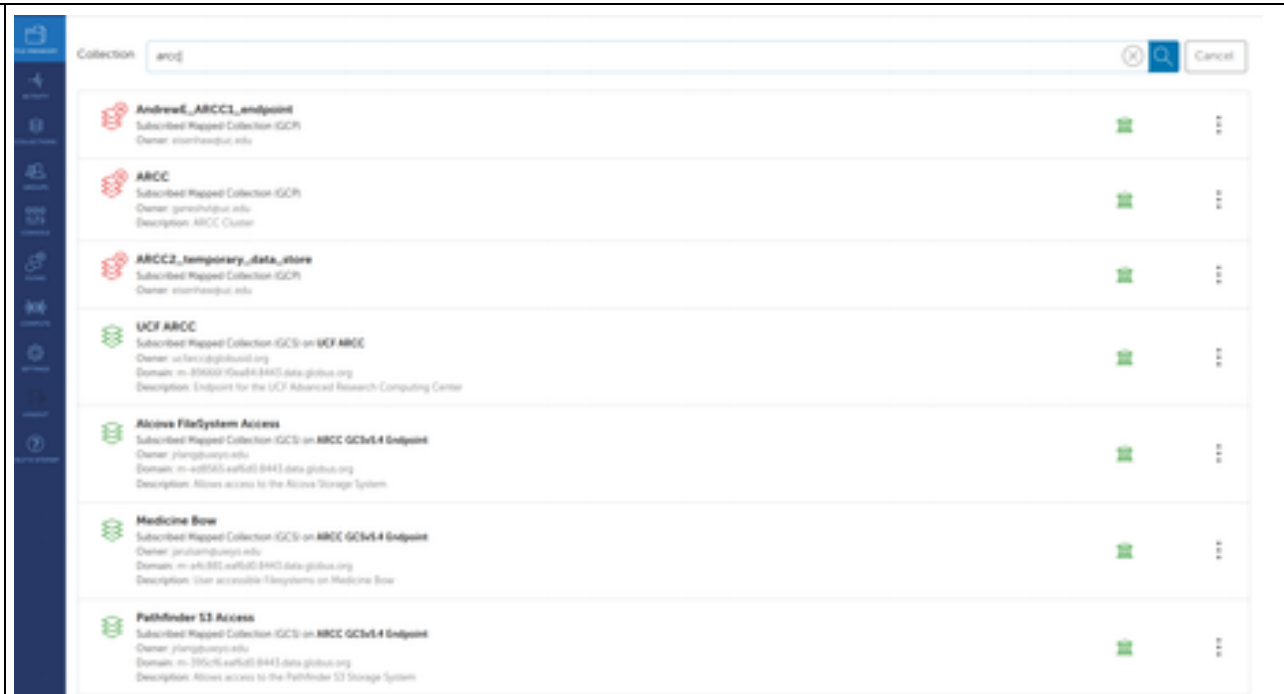
- Start by clicking on the 'search' box to begin looking for the collection you want to use.

The screenshot shows the Globus File Manager interface. On the left is a dark blue sidebar with three menu items: a cloud icon with a 'g' (selected), a folder icon labeled 'FILE MANAGER', and a bookmark icon labeled 'BOOKMARKS'. The main content area has a light gray header with a folder icon and the text 'File Manager'. Below the header, there are two input fields: 'Collection' with a search icon and the word 'Search' in blue, and 'Path' with an empty search box.

ARCC's Collections

By Searching for 'arcc' in the collection search, you can find all collections managed by ARCC. Here you can see the results with different attributes of owner and description. For UWyo ARCC collections, the owner will have a @uwyo.edu email address and a description of the filesystem being used. Here we can find:

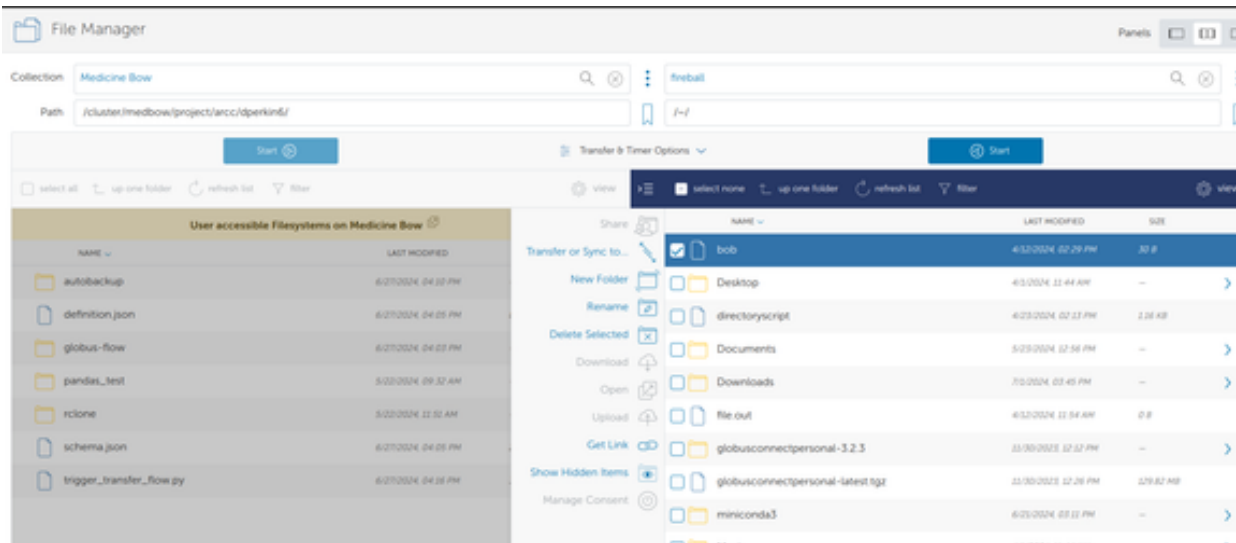
- MedicineBow
- Pathfinder
- And others that may or may not still be active



To access the collections managed by ARCC, you will need to provide your username and password along with your preferred two factor authentication method (Duo Push)

Basic Globus Transfer

Below is an example of a basic transfer from a Globus Personal Endpoint on a laptop to an ARCC system. Essentially after navigating to the directory you want to transfer from and to, select the number of files or directories you want and click “Start”



Enable Globus Personal Endpoint

If you want to enable Globus on your personal or work computer, please follow [the instructions](#) for your Operating System of choice on the Globus website.

Install Globus Connect Personal

Create a Globus collection on your laptop. Globus Connect Personal is available for all major operating systems.



Other Globus Options

This section of the workshop expands upon the previous section on Basic Globus transfers. It is still an introductory level of content, but may not apply to all use cases.

Discussed in this section are the transfer options in Globus, sharing data with others, and then briefly touch on the Globus CLI and flows functionality.

-
- [Globus Transfer Options](#)
 - [Transfer Option Example](#)
 - [Notification and Scheduling Options](#)
 - [Sharing with Globus](#)

- [The Globus CLI](#)
 - [Globus Flows](#)
 - [Next Steps](#)
-

Globus Transfer Options

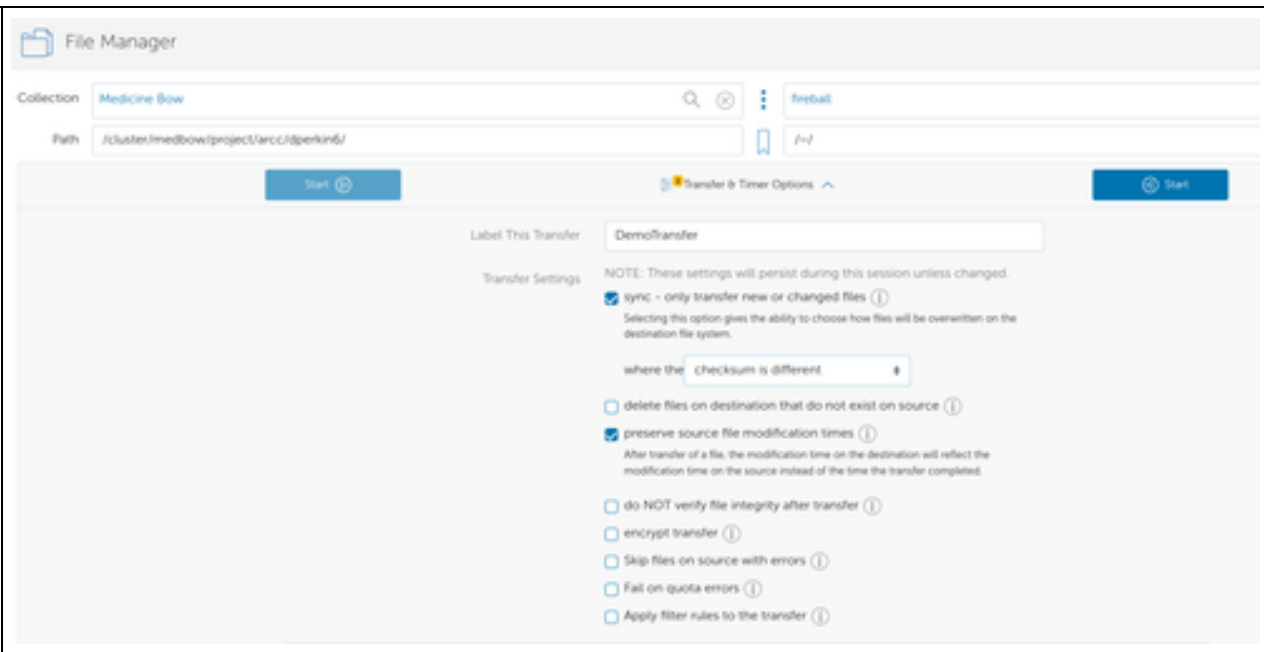
Beyond simply transferring data, Globus enables users optional features that enhance the functionality of data transfers. Here are a few that ARCC frequently uses:

- **Label** - Because Globus keeps a record of each transfer along with the stats about them, it can be hard to determine which transfer is which based solely on the transfer ID and date. Giving a transfer a human-readable label can help users keep track of each transfer.
 - **Sync** - to only transfer new or changed files. Selecting this option gives the ability to choose how files will be overwritten on the destination file system. The options for syncing can be where the modification time of a file is newer, the file size is different, the file on the source does not exist on the destination, or where the checksum is different. *When using this option, ARCC recommends using the checksum option, to cover a variety of scenarios*
 - **Preserve source file modification times** - After transfer of a file, the modification time on the destination will reflect the modification time on the source instead of the time the transfer completed.
-

Transfer Option Example


In this example we expand the 'Transfer and Timer Options' window to provide a demonstration transfer that has a label, syncing, and preserving the file modification time. Notice that there are many more options that can be used as well including:

- Encryption
- Skipping on errors
- Deleting files that don't exist on source



Notification and Scheduling Options

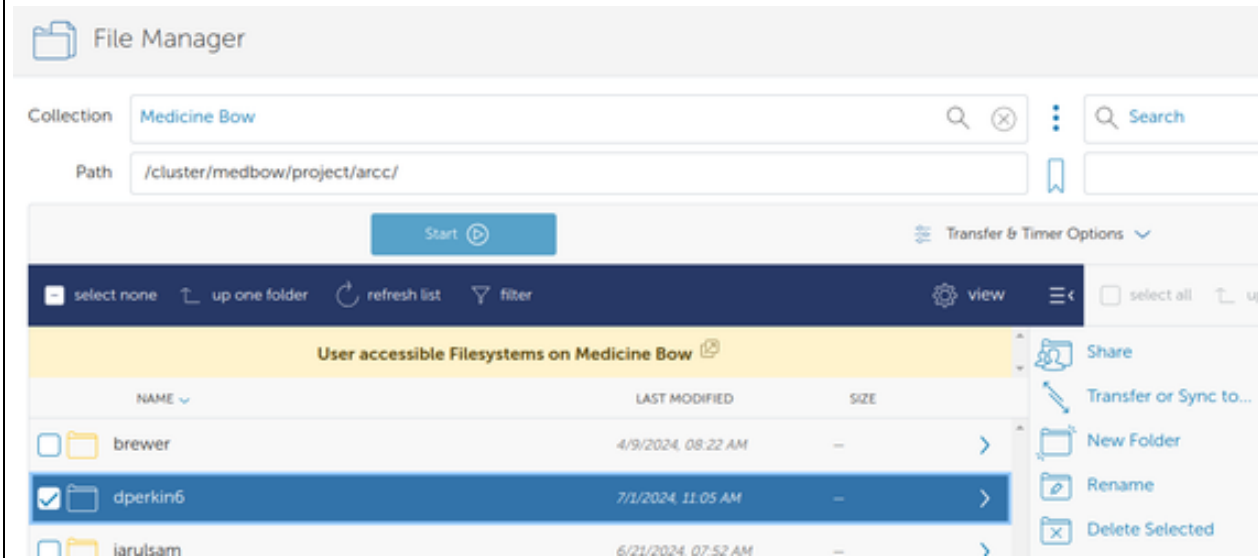
At the bottom of the same Transfer and Timer Options window, there are options to disable notifications of transfers as well as scheduling transfers that may or may not repeat

Notification Settings	<input type="checkbox"/> Disable success notification ⓘ
	<input type="checkbox"/> Disable failure notification ⓘ
	<input type="checkbox"/> Disable inactive notification ⓘ
Schedule Start	<input type="text" value="mm/dd/yyyy, --:-- --"/> ⓘ
 Repeat	<input type="text" value="does not repeat"/> ⌵

Sharing with Globus

Sharing with guest collections is available on Globus endpoints managed by an active Subscription. You can create guest collections on Globus endpoints if you are a member of a Subscription Group.

- By sharing with Globus, the person you are sharing with will need a Globus account and access to an Endpoint
- Please contact ARCC if you are interested in sharing data with Globus



The Globus CLI

Globus does have a command line interface (CLI) that can be used within HPC workflows as well. ARCC does not have this installed by default, but a user can install it using the miniconda module on MedicineBow.

- This still uses the same terminology as basic Globus transfers and users will have to be familiar with those key concepts before attempting to use the CLI

Command Line Interface (CLI)

This page provides information about the Globus CLI, a standalone application that can be installed on the user's machine and used to access the Globus service.

The CLI provides an interface to Globus services from the shell, and is suited to both interactive and simple scripting use cases.

It is open source and available at <https://github.com/globus/globus-cli>

Installation

The Globus CLI is available as a python package from PyPI. It can be installed with `pipx` or `pip`.

We enthusiastically recommend using `pipx` to install the Globus CLI. It provides an isolated environment to keep your CLI installation running smoothly, and it's resilient to changes in your system's python installation.

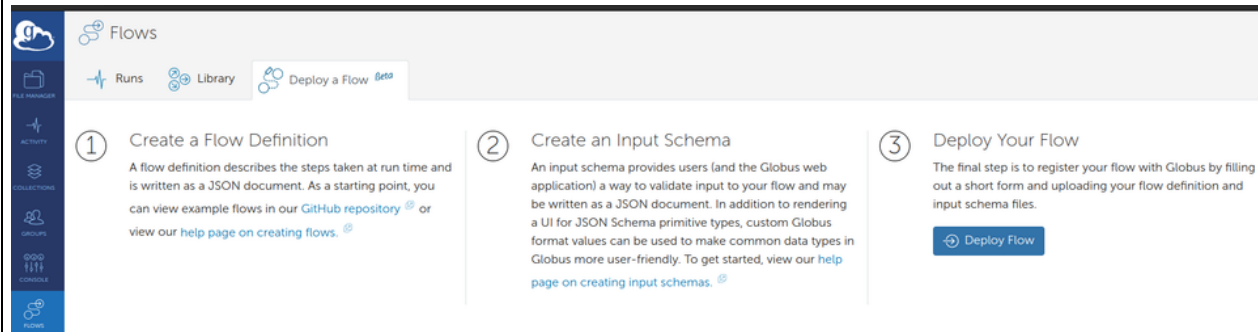


The screenshot shows a web page for installing the Globus CLI. At the top, there are two tabs: 'pipx' (selected) and 'pip'. Below the tabs is a code block containing the command `pip install globus-cli`. To the right of the code block is a 'Copy' button. Below the code block, there is a note that says: 'This is supported anywhere that `pip` works. For example, in an activated `virtualenv`.' Below this note is a blue box with the heading 'Note' and the text: 'We recommend against installing with `pip install --user` or `sudo pip install`. There are known issues with these installation methods.'

Globus Flows

Globus flows enables the automation of data transfers. ARCC is still investigating how this works, but generally speaking, you have to have access to each endpoint you are wanting to transfer to and from and then do three things to deploy a Flow:

- Create a definition with a JSON document
- Create an input schema
- and then deploy the flow



Data Transfer With Desktop Clients

If a web-based application or command line options are not a fit for your workflow, desktop clients are another option to transfer data. While these options are not ARCC provided tools, we have tested many of them and can provide some support for users of them. Clients that ARCC have tested are MobaXTerm, Cyberduck, and Cloudberry (MSP 360 explorer), but for this section of the workshop, we will only use Cyberduck in our examples due to it being available for Windows and Mac operating systems. Before getting into the use of Cyberduck, understanding of transfer protocols is key to the use of these tools and will be discussed first.

- [Transfer Protocols](#)
- [ARCC Tested Data Transfer Clients](#)
- [Connecting to MedicineBow with a Client](#)
- [Logging into MedicineBow With a Client](#)
- [Transferring Data with a Client](#)
- [Next Steps](#)

Transfer Protocols

While the term ‘transfer protocol’ may sound like IT jargon, most people use one everyday. In fact, if you are reading this on your computer you are using the https protocol over the internet to transfer this information onto your browser. There are dozens more transfer protocols, but here we will only talk about some key ones that are used by ARCC everyday.

- ❑ **SFTP** - or the SSH File Transfer Protocol, is an extension of the secure shell protocol that allows for a range of operations on remote files which make it more like a remote file system protocol. SFTP’s extra capabilities include resuming interrupted transfers, directory listings, and remote file removal.
- ❑ **S3** - or Amazon S3, was initially developed for Amazon’s web services, but has been a widely adopted protocol for other companies and services. S3 works for object type storage with terms like buckets, secret key, and access key for accessing and transferring data.
- ❑ **SMB** - or Server Message Block, is a communication protocol developed by Microsoft, that enables file transfers over a shared network. SMB can be used by applications that come default on Windows with File Explorer, Mac Finder, or whatever Linux file navigator you may have.

ARCC Tested Data Transfer Clients

As mentioned above, ARCC has tested the following clients and can provide some support for users of these clients:

- ❑ [MobaXTerm](#) - is a Windows only client that can be used for ssh logins as well as many other protocols, but is also an excellent SFTP client.
- ❑ [Cyberduck](#) - is another client that can be used by Windows and Mac, that we frequently use for working with data on our Pathfinder system due to how well it works with the S3 protocol, but can use many other protocols
- ❑ [Cloudberry](#) - is another Windows and Mac client that can use the S3 protocol, but requires a license to take advantage of it’s full functionality.

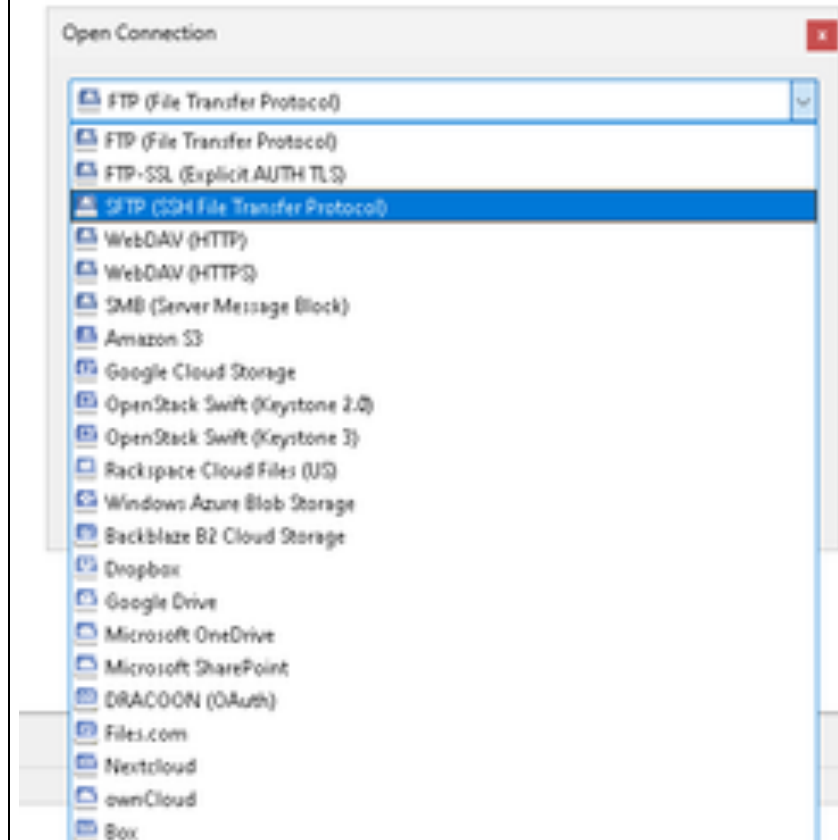
WARNING: You must already have an ARCC account, two factor enabled, and ssh keys or accesskey/secretkey combo prior to attempting to use a desktop client to transfer data!

Connecting to MedicineBow with a Client

You will need to have your ssh keys downloaded prior to connecting

With a Client (Cyberduck in this example) you will want to connect to MedicineBow using the SFTP protocol to transfer data.

- Click the “Open Connection” icon
- Select SFTP (SSH File Transfer Protocol)



Logging into MedicineBow With a Client

You will need to have your ssh keys downloaded prior to connecting

Once SFTP is selected (Cyberduck in this example) you will need to enter the following information:

- Server: medicinebow.arcc.uwyo.edu
- UWYO username
- Password
- Choose the file that is your ssh key. (This example is using Windows and a key tip when looking in your .ssh folder is to select “show all files”

Open Connection

SFTP (SSH File Transfer Protocol)

Server: medicinebow.arcc.uwyo.edu Port: 22

URL: sftp://dperkin6@medicinebow.arcc.uwyo.edu

Username: dperkin6

Password: ●●●●●●●●

Anonymous Login

SSH Private Key: C:\Users\dperkin6\.ssh\id_ecdsa Choose...

Save Password

Connect Cancel

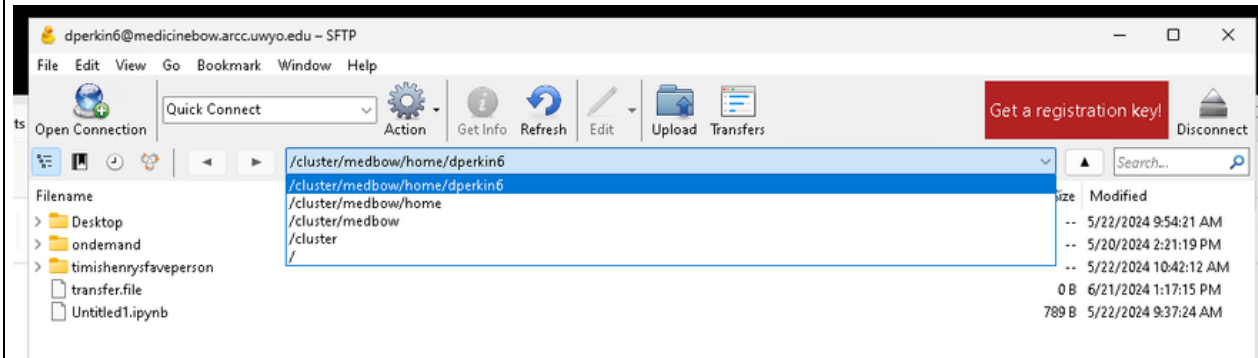
More Options

Transferring Data with a Client

Once SFTP is connected (Cyberduck in this example) you will be connected to your home directory on MedicineBow

- From here you can navigate to anywhere that you have access to.
- To transfer a file you can click the upload icon, or simply “drag and drop” a file from your computer

After hitting “connect”, you will be asked to allow the connection



The process is similar for using Pathfinder, but will use the S3 protocol and accesskey/secretkey combo instead of a username, password, ssh key for MedicineBow.

Command Line Data Transfer

While transferring data over a web application such as Open OnDemand, or over a client such as Cyberduck are easy-to-use, they are difficult to automate within a compute job. However, in some use cases, people may want to transfer data, run some computation on that data, transfer it back and so on. These type of tasks can be accomplished using the command line interface (CLI) on MedicineBow. There are many CLI options to use including the previously discussed Globus CLI, scp, SFTP, and rsync which will all work on MedicineBow, but in this module we will detail rclone because it is ARCC’s recommended command line tools due to it’s ability to work with desktops, HPC, and cloud storage systems as well as it’s ability to be multi-threaded to facilitate faster transfers.

- [scp and SFTP CLI Tools and Examples](#)
- [rsync CLI Tool and example](#)

- [About rclone](#)
 - [rclone Configuration](#)
 - [Using rclone](#)
 - [Next Steps](#)
-

scp and SFTP CLI Tools and Examples

Before diving into providing information on rclone, we'll briefly cover some of the other command line tools and give examples for how to use them on MedicineBow.

- **scp** - Uses SSH (Secure Shell) to authenticate, then securely transfer data. This means the data is authenticated by the user initiating the connection.
 - Example for from local to MedicineBow on Linux/Mac
 - Basic syntax of a scp command: `scp file username@server:directory to transfer to`
 - `dylan@fireball:~$ scp transfer.file dperkin6@medicinebow.arcc.uwyo.edu:/project/arcc`
 - `transfer.file`
 - `100% 0 0.0KB/s 00:00`
 - `dylan@fireball:~$`
 - **SFTP** - can also be used on the CLI as well as clients. Compared to the SCP protocol, which only allows file transfers, the SFTP protocol allows for a wider range of operations on remote files. SFTP clients provide extra capabilities include resuming interrupted transfers, directory listings, and remote file removal.
 - SFTP is generally more platform-independent than SCP.
 - Example of interactive use of SFTP
 - `dylan@fireball:~$ sftp dperkin6@medicinebow.arcc.uwyo.edu`
 - `Connected to medicinebow.arcc.uwyo.edu.`
 - `sftp>`
 - Helpful SFTP commands
 - `?` - is how you access the help, `put` - to upload a file, `mput` - to upload multiple files
 - `get` - to download a file or directory, `mget` - to download multiple files
-

rsync CLI Tool and example

rsync is another very useful tool, that has many options. 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. With the help of rsync command you can copy and synchronize your data remotely and locally across directories, across disks and networks, perform data backups and mirroring between two Linux machines.

- Basic syntax of rsync command: `rsync options source destination`
- `dylan@fireball:~$ rsync transfer.file dperkin6@medicinebow.arcc.uwo.edu:/project/arcc/dperkin6`

Some common options used with rsync commands

```
-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 ownership and timestamps
-z : compress file data
-h : human-readable, output numbers in a human-readable format
```

About rclone

[Rclone](#) is a command-line program to manage files on remote storage. It is a feature-rich alternative to cloud vendors' web storage interfaces. Over 70 cloud storage products support rclone including S3 object stores, business & consumer file storage services, as well as standard transfer protocols. Rclone has powerful cloud equivalents to the unix commands rsync, cp, mv, mount, ls, ncdu, tree, rm, and cat. It is used at the command line, in scripts or via its API.

Rclone mounts any local, cloud or virtual filesystem as a disk on Windows, macOS, linux and FreeBSD, and also serves these over SFTP, HTTP, WebDAV, FTP and DLNA.

Rclone helps you:

- ❑ Backup (and encrypt) files to cloud storage
- ❑ Restore (and decrypt) files from cloud storage
- ❑ Mirror cloud data to other cloud services or locally
- ❑ Migrate data to the cloud, or between cloud storage vendors
- ❑ Mount multiple, encrypted, cached or diverse cloud storage as a disk
- ❑ Union file systems together to present multiple local and/or cloud file systems as one

[Download rclone](#)

rclone Configuration

Rclone does require some configuration for any “transfer partner” this is a long process, but once setup it is useful to use over and over again. Examples will be for configuring transfers to/from MedicineBow using authentication with an ssh-key:

```
dylan@fireball:~$ rclone config
2024/07/10 14:32:44 NOTICE: Config file "/home/dylan/.config/rclone/rclone.conf" not found - using defaults
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> medbow
```

You will then be given choices to make to continue configuring the ‘medbow’ remote configuration, in our case we will pick the number ‘27’ SSH/SFTP Connection

```
Storage> 27
** See help for sftp backend at: https://rclone.org/sftp/ **

SSH host to connect to
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
1 / Connect to example.com
```

```
\ "example.com"  
host> medicinebow.arcc.uwyo.edu  
SSH username, leave blank for current username, dylan  
Enter a string value. Press Enter for the default ("").  
user> dperkin6  
SSH port, leave blank to use default (22)  
Enter a string value. Press Enter for the default ("").  
port>
```

Then you will have to give more login info, in this case we will accept defaults by hitting ‘enter/return’ to continue until we get to the ‘key_file’ option and enter the location of the ssh key file:

```
SSH password, leave blank to use ssh-agent.  
y) Yes type in my own password  
g) Generate random password  
n) No leave this optional password blank (default)  
y/g/n>  
Raw PEM-encoded private key, If specified, will override key_file parameter.  
Enter a string value. Press Enter for the default ("").  
key_pem>  
Path to PEM-encoded private key file, leave blank or set key-use-agent to use ssh-agent.  
  
Leading `~` will be expanded in the file name as will environment variables such as `${RCLONE_CONFIG_DIR}`.  
  
Enter a string value. Press Enter for the default ("").  
key_file> /home/dylan/.ssh/id_ecdsa
```

The next options will be relating to passwords and certificates. For MedicineBow, none of this applies so we keep hitting enter until we get to the cipher where we will enter '1' for false. The final two steps are entering an advanced configuration and then saving before exiting the configuration setup.

The passphrase to decrypt the PEM-encoded private key file.

```
Only PEM encrypted key files (old OpenSSH format) are supported. Encrypted keys  
in the new OpenSSH format can't be used.  
y) Yes type in my own password  
g) Generate random password  
n) No leave this optional password blank (default)  
y/g/n>
```

```
Choose a number from below, or type in your own value
 1 / Use default Cipher list.
   \ "false"
 2 / Enables the use of the aes128-cbc cipher and diffie-hellman-group-exchange-sha256, diffie-hellman-
group-exchange-sha1 key exchange.
   \ "true"
use_insecure_cipher> 1
Disable the execution of SSH commands to determine if remote file hashing is available.
Leave blank or set to false to enable hashing (recommended), set to true to disable hashing.
Enter a boolean value (true or false). Press Enter for the default ("false").
disable_hashcheck>
Edit advanced config? (y/n)
y) Yes
n) No (default)
y/n> n
Remote config
-----
[medbow]
host = medicinebow.arcc.uwyo.edu
user = dperkin6
key_file = /home/dylan/.ssh/id_rsa
use_insecure_cipher = false
-----
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d> y
```

Using rclone

The basic syntax goes as follows `rclone <function> <source> <destination endpoint>:<bucket>`.

the basic functions are:

- copy** - to copy files/directories to or from somewhere
- sync** - (one way) to make a directory identical

- ❑ **move** - files to cloud storage deleting the local after verification
- ❑ **check** - for missing/extra files
- ❑ **mount** - your cloud storage as a network disk

More information on each function can be found at <https://rclone.org/#what>. An example of a copy from local to MedicineBow would be:

```
dylan@fireball:~$ rclone copy transfer.file medbow:/project/arcc
```

Mapping/Mounting with SMB

As mentioned previously, data transfer protocols impact how to connect to remote storage and methods of transfer. SMB or Server Message Block by Microsoft is another one of these methods. There are multiple differences about it from the other methods, but most important it is very easy. It allows systems on the same network to share files between computers as though the files were on the computer's local hard drive. Closest to file moving interfaces most familiar to many on Mac/PCs (with drag and drop or cmd/ctrl+c, cmd/ctrl+p). Discussed in this section of the workshop are the advantages and disadvantages and how to use SMB on ARCC systems.

- ❑ [Advantages of SMB](#)
 - ❑ [Disadvantages of SMB](#)
 - ❑ [How to Use SMB on Windows](#)
 - ❑ [Map a Network Drive on Windows](#)
 - ❑ [Connect to the Remote Storage \(Windows\)](#)
 - ❑ [How to Use SMB on Mac](#)
 - ❑ [Connect to Remote Storage \(Mac\)](#)
 - ❑ [Next Steps](#)
-

Advantages of SMB

As mentioned previously, the ease of use is the key feature of SMB protocol, but there are indeed more that warrant it's use.

1. **Interactive use of remotely stored files** - once connected to the remote storage, a user can open, view, and edit files without actually transferring the files themselves. All changes are saved on the remote storage. All this is dependent on the size and type of files, but many are supported including spreadsheets, word processing, code, images, and even video files.
 2. **Re-connecting** - once the storage is mapped/mounted, the option to save the username and password for the connection exists. Meaning all one has to do to re-connect to the remote server is to click on the connection to get to the storage.
 3. **In-network sharing** - on ARCC systems, anyone who has a UWyo username and password that can connect to the campus VPN, can be given permission to any storage that can be connected with SMB.
 4. **Protected connections** - once connected using SMB, the connection itself is encrypted. Meaning it is very difficult for a malicious actor to exploit any open connections a user has.
-

Disadvantages of SMB

Despite all of the good things SMB connections provide, there are a few problems that can arise.

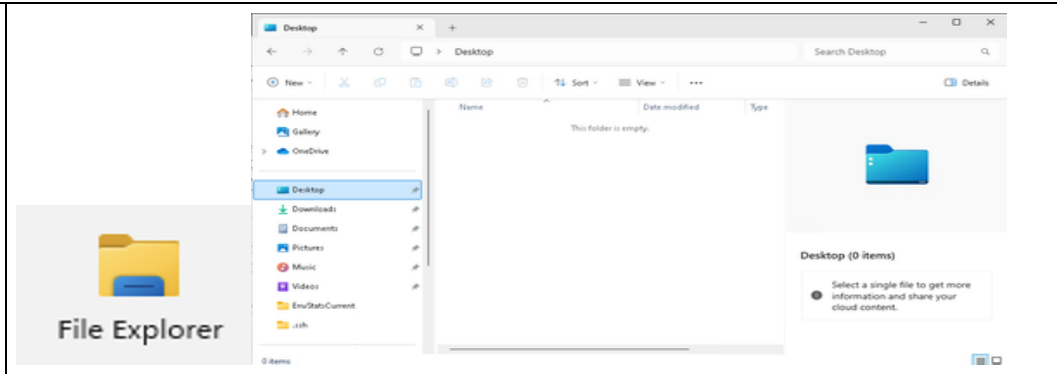
1. **No out-of-network access** - on ARCC systems, users are required to have a UWyo account and have the ability to log into the network such as the UWyo wifi or VPN. Meaning if researchers want to collaborate with others from different institutions AND wishing to use SMB to share data, additional steps must be taken to enable this and has an associated cost.
 2. **Performance** - SMB is not a multi-threaded protocol, like other protocols previously discussed in this tutorial and transfers can be painfully slow.
 3. **Potential corruptions** - SMB does not do any file verification when transferring data such as checksums. This disadvantage can become problematic should a network error, power outage, or other interruption occur.
 4. **No automation** - a major disadvantage to using SMB is that the protocol cannot be used within an HPC job to transfer data between systems.
-

How to Use SMB on Windows

Using the SMB protocol on Windows is pretty simple as long as the initial requirements of having a UWyo account and access to a UWyo network are met along with the permission of a UWyo based project PI.

The first step in connecting is to open the Windows File Explorer application by clicking on the folder icon in the start menu.

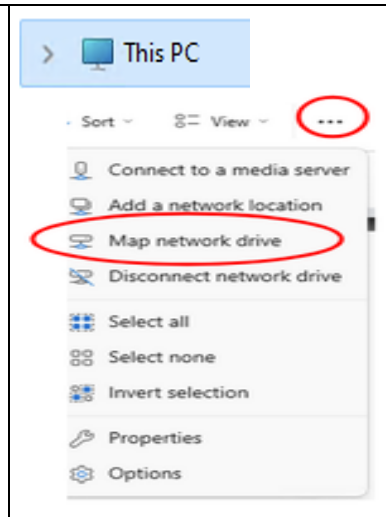
Once the application is open it should look like the file browser that you are familiar with.



Map a Network Drive on Windows

Next take the following steps to map a network drive:

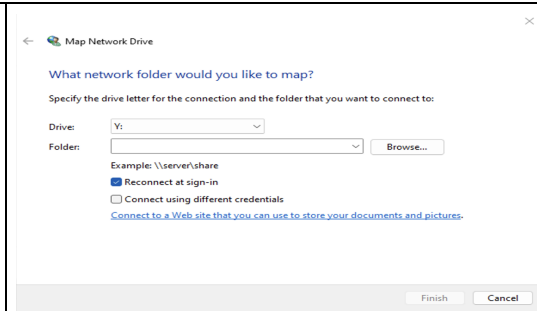
- Find the section on the left side of the File Browser that says “This PC”
- Select that option
- Once there, on the menu on the top there will be three dots (...):
 - This brings up a drop down menu
 - Select Map a network drive



Connect to the Remote Storage (Windows)

Once the prompt for “What network folder would you like to map?” appears, you can then add the path to the ARCC system you want to connect to.

- The ARCC Data Portal path is:
\\data.arcc.uwyo.edu\cluster\alcova
- It is a good idea to check the box that says “Reconnect at sign-in” to keep your connection active whenever you use this computer
- If the computer you are using is not a UWyo-owned computer, you may have a different username on that computer. If that is the case, check the box that says “Connect using different credentials”



If you have a local user, you will have to use your UWyo username and password to connect to the system.

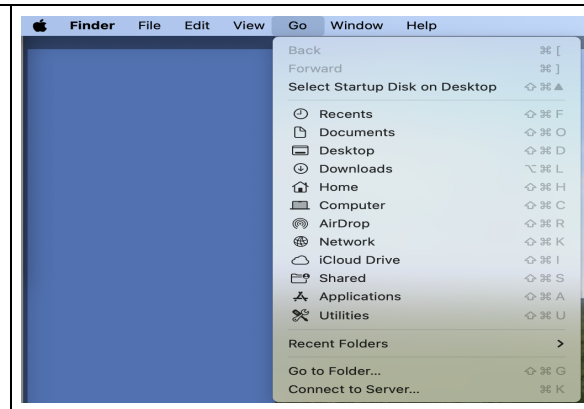
This example is using the letter ‘Y’, you can choose any letter other than ‘C’

How to Use SMB on Mac

The process is similar on Mac, but just a little different

The first step is to open the Mac Finder application, once there find the option for “Go” on the top menu.

- This opens a drop down menu
- Select the option for “Connect to Server”



Connect to Remote Storage (Mac)

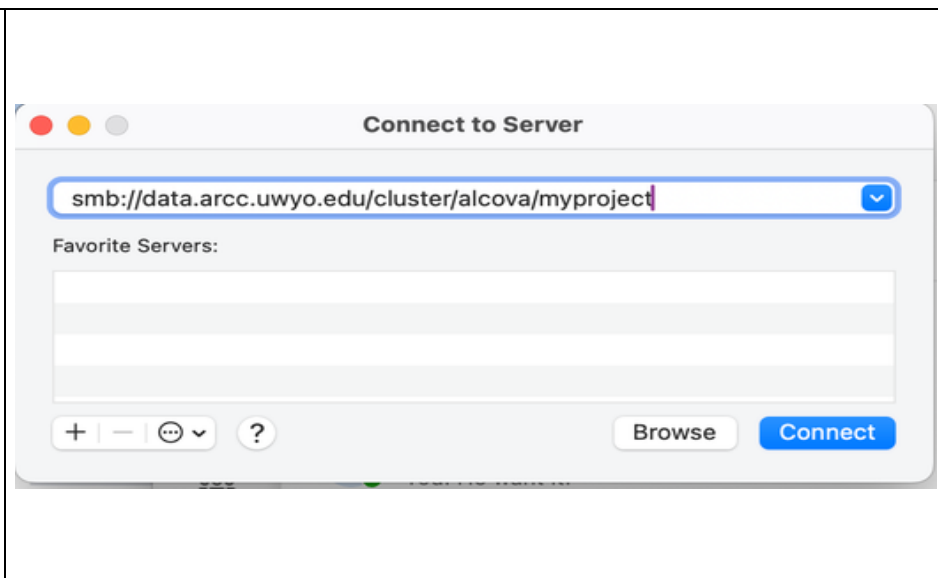
When the “Connect to Server” prompt appears you will need to enter the address prefixed with ‘smb’

For the ARCC Data Portal the address is:

smb://data.arcc.uwyo.edu/cluster/alcova

- Then click ‘connect’ button.
- When prompted for username and password:
 - Enter your UWYO username
as: *uwyo\username*

Enter your UWYO password
like you would normally



Next Steps

Previous Command Line Data Transfer	Workshop Home Intro to Data Transfer with ARCC
---	--