Data Literacy and Data Management

HPRC Short Course
September 2018

Rick McMullen, Ph.D.

Associate Director, HPRC
Texas A&M University
mcmullen@tamu.edu

For More Help…

Website:	hprc.tamu.edu
Email:	help@hprc.tamu.edu
Telephone:	(979) 845-0219
Visit us in person:	Henderson Hall, Room 114A

Help us, help you -- we need more info

- Which Cluster
- UserID/NetID
- Job id(s) if any
- Location of your jobfile, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem

Goals for the session – What are yours?

- Present a conceptual framework for the life cycle of data
- Present a case for attending to managing your data in an organized way
- Learn about the concept of the “life-cycle” of data
- Learn about some tools and systems for managing your data
    - Storing
    - Organizing and finding
    - Moving

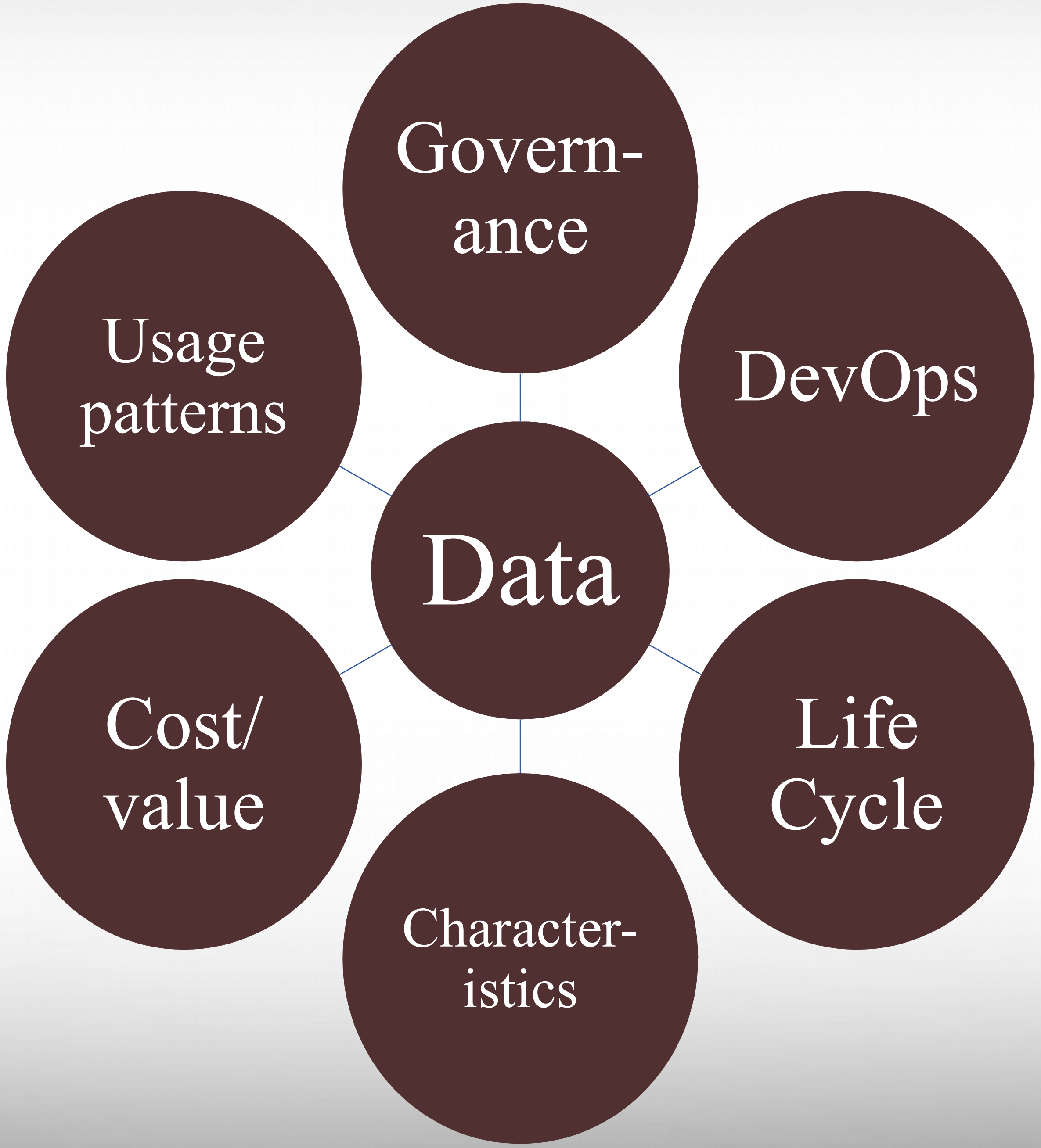
Outline

- Framework issues
  - Awareness about your data/information assets
  - Conceptual framework for the ”life cycle” of your data
- Some specific considerations
  - Kinds of data you might use in your research or scholarship
    - What types do you use now?
  - Storing your stuff
    - Ex. 1: Working with files in Linux
    - Ex. 2: Organizing using directories
    - Ex. 3: Creating and using archives with *tar* and *zip*
  - Finding your stuff (metadata, organization and searching)
    - Ex. 4: Using *grep* to search within files
    - Ex. 5: Using *find* to locate files
  - Moving your data from here to there
  - Using git and github as a way to organize, manage and share data

Questions to think about

- What is your work about?
- Who do you work with? Within your facility? National? International?
- What kind of data do you work with and where does it come from?
- Where do you do your computing? What resources do you use now? Is there a data or computing coordination center?
- What bottlenecks and issues have you identified? Are these process or infrastructure related?

Lenses for looking at your information assets



Data

Governance

DevOps

Life Cycle

Characteristics

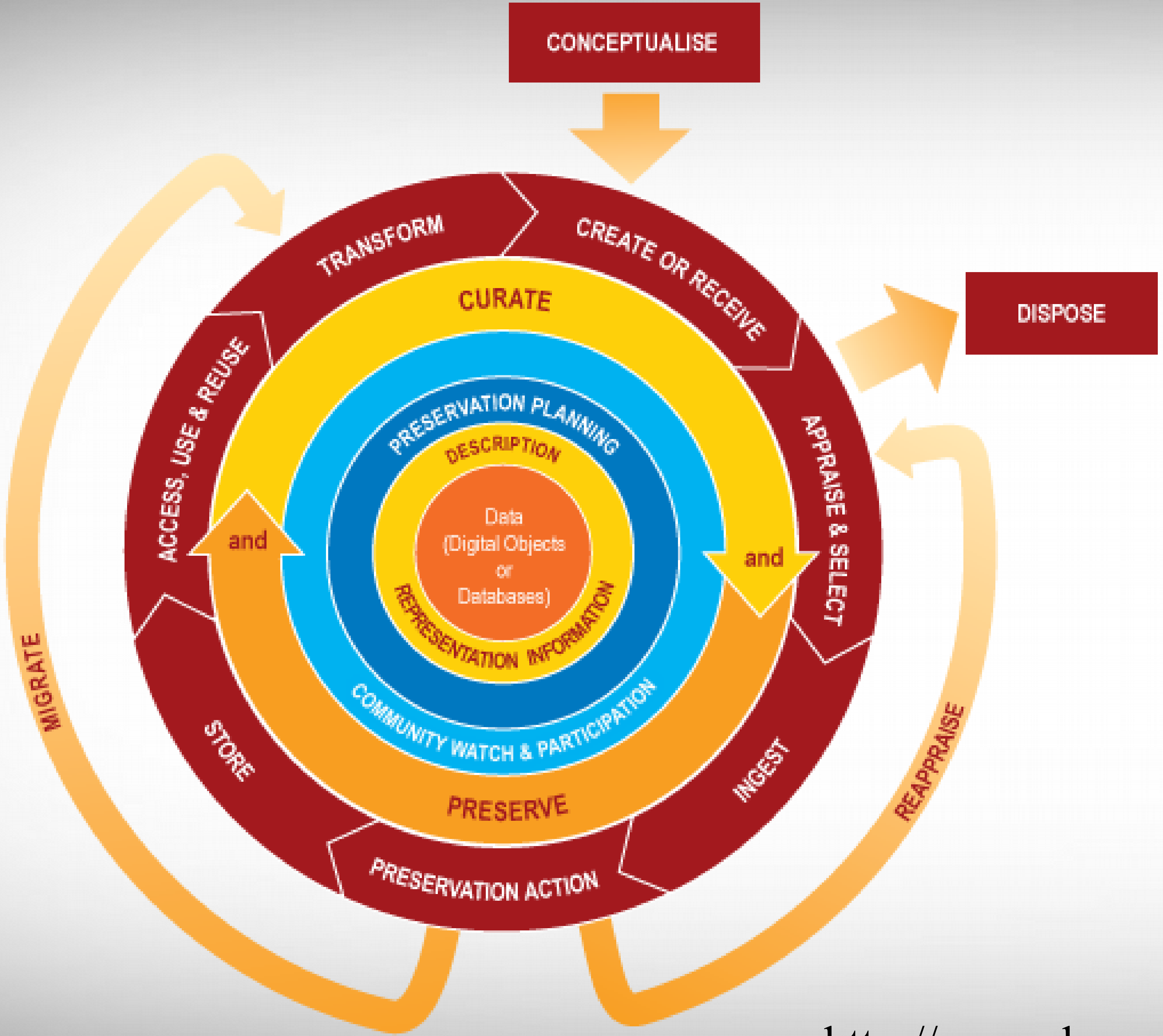
Cost/ value

Usage patterns

It’s Your Data! What’s at risk?

- Your Dissertation or Thesis
- Your research grant
- Your research collaboration and collaborators
- Maintaining compliance with laws, regulations, policies
- Your personal and professional reputation!

Conceptual framework for the life-cycle of data



Key elements of the Data Curation Centre Curation Lifecycle Model

1. Conceptualise
2. Create or Receive
3. Appraise and Select
4. Ingest
5. Preservation Action
6. Store
7. Access, Use and Reuse
8. Transform

http://www.dcc.ac.uk/resources/curation-lifecycle-model/

Let’s focus on these for now

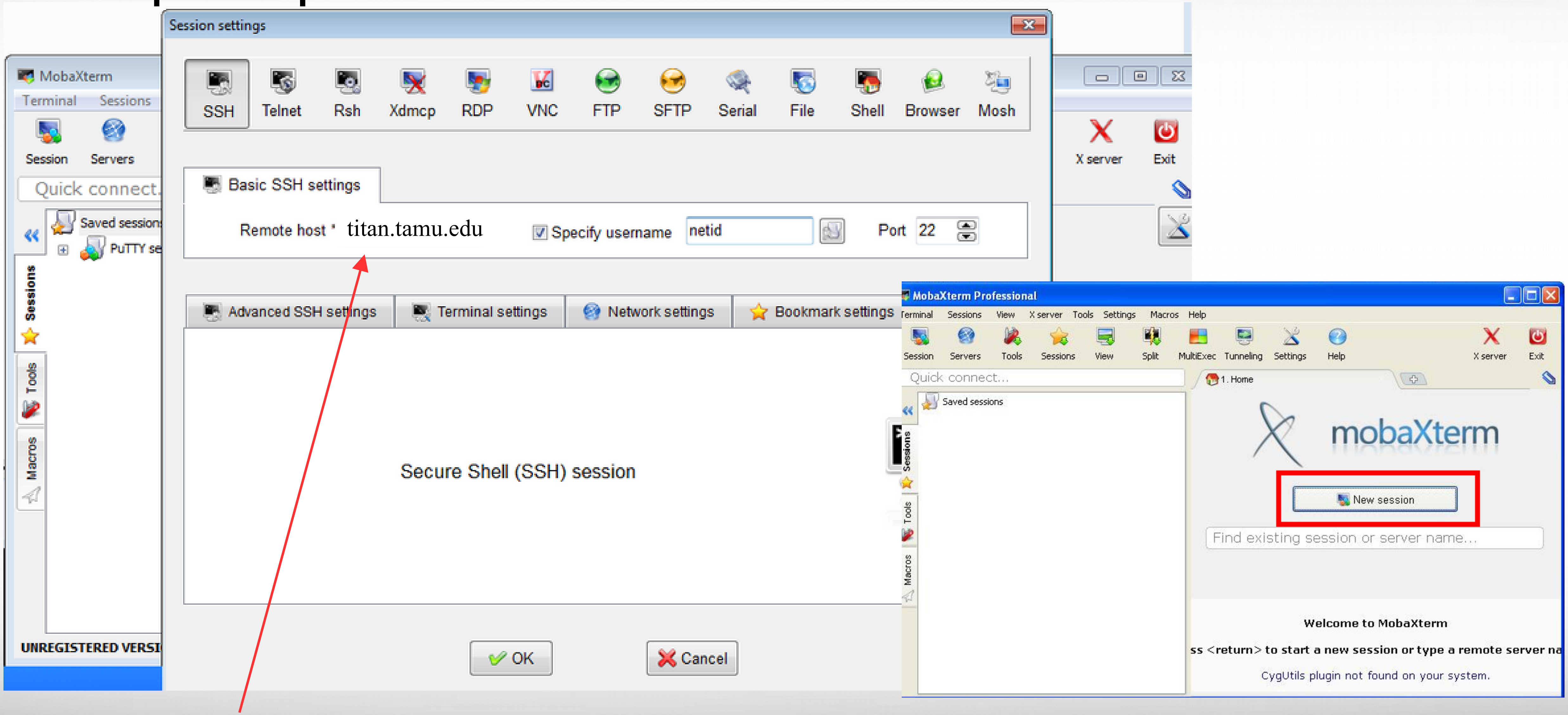
- Storing your data
- Annotating and finding your data
- Moving your data

Exercise 0: Getting logged in to the class machine

- IFF you signed up for this class through the HPRC Short Course web pages THEN you should have a log-in on the class machine
- Recommended: Log in to the lab PC, then use MobaXTerm Personal Edition to log in to the class machine `titan.tamu.edu` using your TAMU NetID and password. (Session -> SSH)
  - You *may* need to create C:\Users\<you>\AppData\Roaming\MobaXterm
- Using your own laptop: <u>On campus or connected through the TAMU VPN</u>, connect to `titan.tamu.edu` using MobaXTerm or some combination of X server and ssh.
  - `Eg. ssh -Y -l <myNetID> crystal.tamu.edu`

Using SSH - MobaXterm (on Windows)

https://hprc.tamu.edu/wiki/HPRC:MobaXterm



Session settings

Basic SSH settings

Remote host * titan.tamu.edu

☑ Specify username netid

Port 22

Secure Shell (SSH) session

Welcome to MobaXterm

ss <return> to start a new session or type a remote server na

CygUtils plugin not found on your system.

Use **titan.tamu.edu** as Remote host name.

Using SSH (on a Linux/Unix/OS X Client)

```
ssh –Y NetID@titan.tamu.edu
```

ssh means “secure shell”; -Y means “forward X11 graphics to me”; the rest is the machine you want to log into and your username there

You may see something like the following the first time you connect to the remote machine from your local machine:

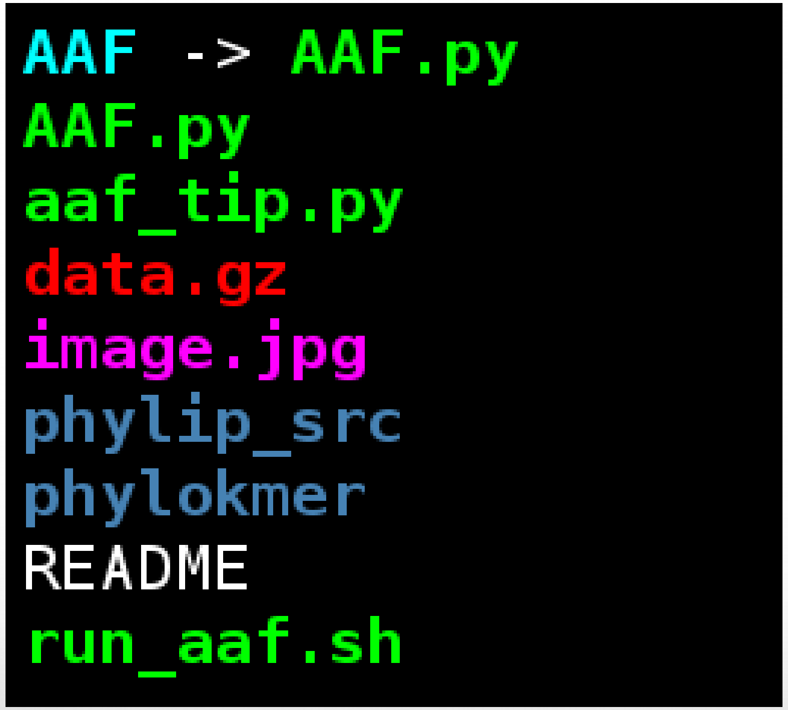
```
Host key not found from the list of known hosts.
     Are you sure you want to continue connecting (yes/no)?
```

Type **yes**, hit enter and you will then see the following:

```
Host ’titan.tamu.edu' added to the list of known hosts.
NetID@crystal.tamu.edu's password:
```

UNIX Terminal Attributes

File and directory names are colored
based on their attributes such as permissions and extension



```
AAF -> AAF.py
AAF.py
aaf_tip.py
data.gz
image.jpg
phylip_src
phylokmer
README
run_aaf.sh
```

TURQUOISE	Symbolic link
GREEN	Executable file
RED	Compressed files
PURPLE	Image files
BLUE	Directories
WHITE	Text files

Basic Linux Shell Commands Review (right?)

- Where am I? Print working directory: `pwd`
- Change directory: `cd <new directory path>`
- List directory: `ls`
- List directory with details: `ls –l`
- Create new directory: `mkdir <directory path>`
- Remove a file: `rm <file path>`
- Remove a directory: `rmdir <directory path>`
- Remove a directory and all its files and directories: `rm –r <directory path>`
- Copy a file: `cp <source path> <target path>`
- Move a file: `mv <source path> <target path>`

Break for 15 minutes

Storing your data

Ways to store and organize your data

- Spreadsheets: Excel, OpenOffice, Google Sheets
  - Convenient, ubiquitous, easy to use
  - Hard to search for specific items (rows) in a large number of spreadsheets
- Databases: “local” SQL, SQL server, NoSQL (key-value, tuple, etc.)
  - Easy to share and keep consistent
  - Someone has to be the database manager
  - Google Datastore, BigTable, Cloud SQL, Azure Data Market databases
- Files and directories -> tar archives
- Structured file formats (e.g. XML, JSON, discipline or vendor-specific)
- Cloud services - AWS, Azure, Google, GitHub; c.f. `https://github.com/sr320/LabDocs/wiki/Data-Management`

Places to store your data

- Bad
  - Not-backed-up laptop or desktop
  - RW-CDs and DVDs (bit rot, labeling, filing and tracking)
  - USB external hard drives (usually consumer grade, no backup, easy to drop)
  - USB thumb drives (can be damaged by handling)
  - Your laptop (drop --> damaged hard drive)
- Good
  - File server with RAID and backup (https://en.wikipedia.org/wiki/RAID)
  - Backed-up laptop or desktop as long as it is backed up frequently
  - Cloud storage (Google, Amazon, Azure, GitHub, DropBox) as long as you understand the risk implied in the Service Level Agreement, and if you keep multiple copies
  - Managed database server (also with regular exports if possible for backup)

Cloud storage…

- Important for data acquisition and exchange from multiple sites in a research collaboration
- Security is really not a problem *at the cloud vendor's end*. Check AWS's compliance and assurance program page: https://aws.amazon.com/compliance/
- Governance needs to cover all data, even in the cloud, and the economics are different to on-prem
    - Capacity is an ongoing cost
    - No depreciation of infrastructure
    - Metadata may be more difficult to collect, making curation more difficult
    - Life cycle considerations are more concrete – doing nothing costs $$ in real time.

Other considerations

- Service labs generally won’t save your data for any length of time (e.g. Microscopy, XRF, NMR, Mass spec…)
- Other than file-format-specific metadata, it’s up to you to organize your stuff appropriately
- Cloud services are OK as long as you understand the risks, limits and financial aspects
- Many disciplines offer repositories for specific kinds of data
  - https://www.nature.com/sdata/policies/repositories
- NSF, NIH, NASA and NOAA require a data management plan for ongoing access to publicly funded research data.

Archiving data

- For moving lots of files, backup and “cold storage”
- Reduces many files and directories to one file
- An archive is easier to handle than a directory hierarchy full of files
- Popular archive tools and formats on all platforms:
  - tar
  - zip
  - cpio

The **tar** Command - examples

```
cd
```

go to your home directory

Package the temp directory into a file called my_hg19.tar

```
tar -cvf my_hg19.tar temp
```

Package the temp directory into a compressed file called my_hg19.tar.gz

```
tar -cvzf my_hg19.tar.gz temp
```

Show the contents of the compressed tar file

```
tar -tzf my_hg19.tar.gz
```

Change the name of your original temp directory so you don’t overwrite it

```
mv temp temp_orig
```

Extract all contents from the compressed tar file

```
tar -xvzf my_hg19.tar.gz
```

-c	=	create
-f	=	file
-t	=	list contents
-v	=	verbose
-z	=	gzipped
-x	=	extract

zip - Archive and compress

Create an archive:

```
$ zip -r archive5.zip test-dir
```

Check the contents:

```
$ unzip -l archive5.zip
```

Extract it:

```
$ cd /tmp/extract-dir
$ unzip /tmp/archive5.zip
```

Exercise 1: Working with files in Linux

- Get a copy of the sample data in your own directory

```
•  cd ~              # go to your home directory
                     # copy recursively from mine
•  cp -R ~mcmullen/sample_data . # that is a dot at the end
•  ls –lR sample_data          # see what’s there
```

- The directory has several sub-directories and different types of files
  - Data I collected for a research project, text and image files
  - Text files are mostly CSV spreadsheets
  - Folders are labeled and grouped sensibly

Exercise 2: Organizing using directories

- Create a subset of the Air Quality data

  -
    ```
    cd ~             # go to your home directory
    mkdir –p new_study/AQ20180119 # make a directory for new work
    ```

  -
    ```
    cp -R sample_data/*.csv    new_study/AQ20180119
    ```

  -
    ```
    ls –lR new_study         # see what’s there
    ```

- Find a file with data about the year 2000

  -
    ```
    cd new_study/AQ20180119
    ```

  -
    ```
    grep 2000 *
    ```

  -
    ```
    cd ~
    ```

  -
    ```
    find new_study –exec grep 2000 {} \; -print
    ```

Exercise 3a: Creating and using archives with tar

- Create a tar archive:
  - cd ~   # go to your home directory
  - # create a tar archive named **sample_data.tar**
  - # from the directory **sample_data**
  - tar cvf sample_data.tar sample_data
  - ls -l   # check the results

- Move it somewhere and ”reconstitute” it
  - mkdir /tmp/mystuff ; cp sample_data.tar /tmp/mystuff
  - cd /tmp/mystuff ; ls –l   #see if the tar file is there
  - tar xvf sample_data.tar   # extract it here in /tmp/mystuff
  - ls –lR   # check the results

Exercise 3b: Creating and using archives with zip

- Create a zip archive:
  - `cd ~          # go to your home directory`
  - `# create a zip archive named` **`sample_data.zip`**
  - `# from the directory` **`sample_data`**
  - `zip -r sample_data.zip sample_data`
  - `ls -l          # check the results`

- Move it somewhere and ”reconstitute” it
  - `cp sample_data.zip /tmp/mystuff  ; rm –rf sample_data`
  - `cd /tmp/mystuff  ;  ls –l  #see if the tar file is there`
  - `unzip sample_data.zip    # extract it here in /tmp/mystuff`
  - `ls –lR           # check the results`

Exercise 4: Using grep to search within files

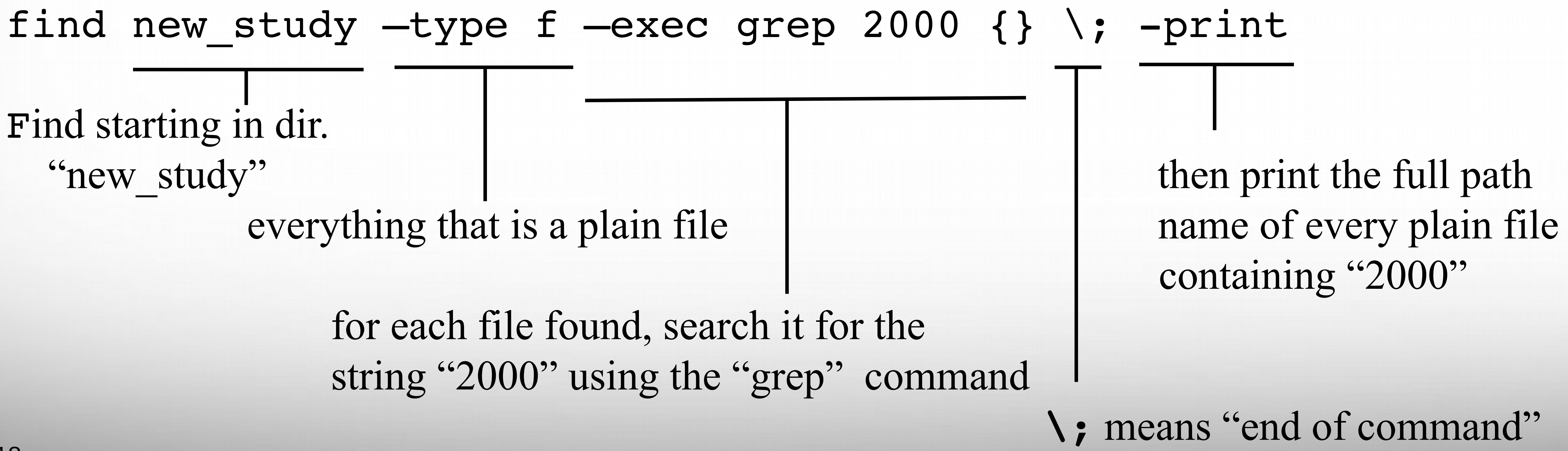
- Use grep to find text strings in files
- Find a file with data about the year 2000
  - `cd new_study/AQ20180119`
  - `grep 2000 *`
  - `cd ~`
  - `find new_study —exec grep 2000 {} \; -print`

Exercise 5: Using `find` and `grep` to find text across multiple directories

- `find` traverses a directory structure and executes a command on each file that matches some specification, generally all files and directories
- Find a file with data about the year 2000 in directory new_study
  - `cd ~       # go home`
  - `#find everything in “new_study”`
  - `find new_study -print`
  - `# starting at directory ”new_study” check all files for “2000”`
  - `find new_study –type f –exec grep 2000 {} \; -print`

- Find a file with data about the year 2000 in directory new_study
  - `cd ~       # go home`
  - `#find everything in “new_study”`
  - `find new_study -print`
  - `# starting at directory ”new_study” check all files for “2000”`
  - `find new_study –type f –exec grep 2000 {} \; -print`

Translation:



```
find new_study –type f –exec grep 2000 {} \; -print
```

Find starting in dir. “new_study”

everything that is a plain file

for each file found, search it for the string “2000” using the “grep” command

`\;` means “end of command”

then print the full path name of every plain file containing “2000”

Grep, compression and zip

- Tar can also compress/decompress the archives it creates
  - `tar –c –z –v -f my_archive.tar.gz <source>`
  - `tar –x –z –v -f my_archive.tar.gz`

  - `-c = create archive;`
  - `-x = extract files;`
  - `-z = gzip it;`
  - `-f = file name to create or extract from`
  - `<source> is usually a directory path name`

- `.tar.gz == .tgz`
- the suffix doesn't matter, tar looks at the “-z” switch

Compression efficiency

```
$ du -s -h sample_data
117M sample_data
$ ls –l -h sample_data.*
-rw-rw-r-- 1 mcmullen mcmullen 45M Jan 19 11:20 sample_data.tar.gz
-rw-rw-r-- 1 mcmullen mcmullen 45M Jan 19 11:20 sample_data.zip
```

- 45/117 = 38% of the original
- gzip and zip give comparable results for this data
- Try zip and gzip with ~mcmullen/AMillionRandomDigits.bin

Data Cleaning

- Processing pipelines and batch scripts (these preserve methodologies and make them reusable)
- OpenRefine (openrefine.org) for cleaning tabular and relational data

OPEN
Refine
A free, open source, powerful tool
for working with messy data

  and performing extract-transform-load (ETL) tasks.
  Also Talend Open Studio (https://www.talend.com/download/talend-open-studio/)

- Save everything!

Transformations are computations, VMs are research data

- The computing you do on your data are as important as your data WRT reproducibility and documentation of method
- It is becoming possible and useful to capture your data processing environments and computational research tools as Virtual Machines
- VMs can be saved in a ”portable” format, Open Virtualization Format
- End-user VM systems: VirtualBox, KVM/qemu (free), VMWare (cost)
- Also, “containers” are a more lightweight option for preserving complete work environments
  - Docker (not “dockers” ;) - https://www.docker.com/
  - Singularity (not the Kurzweil thing ;;) https://singularity.lbl.gov/ , https://www.sylabs.io/docs/

VM Service Examples

- Obviously Amazon Web Services, Google, Azure etc.
- More focused on science and research:
  - Jetstream (NSF) - https://jetstream-cloud.org/
    - Jetstream allows VMs to be catalogued as publications (get a doi, keep in a public repository)
  - CyVerse - http://www.cyverse.org/, “Cyberinfrastructure for Data Management and Analysis”, mostly about biology and bioinformatics

Metadata

Annotating your stuff - Metadata

- “Data about data” – who, what, when, why, how
- ***Critically important if you want to find anything and understand what it means more than a week from now***
- Directory and file naming schemes
- Internal metadata (e.g. TIFF image headers)
- Spreadsheet column names
- Database data directories and field names
- XML Schema (tags and optional values)
- Disk labels, textual documentation

File types are metadata!

- One kind of typing is a file suffix, e.g. .c, .xlsx, .sql
  - Not 100% reliable as you can rename a file to anything
- Files also have internal, sometimes characteristic typing information
  - Mime types
  - Magic numbers
- The Linux “file” command can tell you a lot about a file

Exercise – use “file” command

```
$ cd ~
# go to the landsat imagery directory
$ cd sample_data/landsat\ imagery
$ ls -l
-rw-r--r-- 1 mcmullen mcmullen 90738310 Jan 18  2018 Landsat_scene_list_index.csv
drwxr-xr-x 2 mcmullen mcmullen     4096 Sep 13 13:05 p001r27_1m19760327.MSS-EarthSat-
Orthorectified.edu

$ file Landsat_scene_list_index.csv
Landsat_scene_list_index.csv: ASCII text

$ cd p001r27_1m19760327.MSS-EarthSat-Orthorectified.edu/
$ file p001r27_1m19760327_01.tif.gz
p001r27_1m19760327_01.tif.gz: gzip compressed data, was
"p001r27_1m19760327_01.tif",
from Unix, last modified: Wed Apr  4 20:46:10 2001
```

File hierarchy and “name” metadata

```
Growth_rates_enz_1            - Directory
   Read.me         - File with description of method
   Experiment_1          - Directory
      Image_0001_date_time     - File with observations
      …
      Image_9999_date_time     - File with observations
   Experiment_2
   …
```

- Hard to change your mind if you need to modify your metadata schema, e.g. add a location where the experiment took place
- Easy to bundle and export your data at any level of the file tree using Unix “tar” command

Finding your stuff - searching

- Search for names and types of files
  - Linux/Unix/Mac OS X: “find” command
  - GUI file browser search
- Search for text or text patterns in files
  - Linux/Unix/Mac OS X: “grep” command in a directory
  - Within specific named files: “find … -exec grep … {} \; -print”
- Spreadsheet search box
- Databases – SQL/noSQL queries,
- Web-based information – Google site search, Microformats

Large scale data management issues

- Getting the right storage system or service for the volume, variety, and velocity of your data
- Tools for automating tasks (metadata extraction, cataloging, tiered storage management)
- Managing risk: security and compliance concerns (e.g. HIPAA, FERPA, licensed data with restrictions and terms, etc.)

A couple of metadata tools for “big” collections

- Robinhood Policy Engine - https://github.com/cea-hpc/robinhood/wiki
  - Policy Engine: schedule actions on filesystem entries according to admin-defined criteria, based on entry attributes.
  - User/group usage accounting, including file size profiling.
  - Extra-fast 'du' and 'find' clones.
  - Customizable alerts on filesystem entries.
  - Aware of Lustre OSTs and pools.
  - Filesystem disaster recovery tools.
  - Open, LGPL-compatible license.
- Starfish Storage - http://www.starfishstorage.com/
  - Similar to Robinhood but supports cloud-based storage as well as local POSIX FSs
  - Not free.

Bigger picture questions to think about

- Do you have a business case for using cloud vendors for R&D computing/storage tasks?
- What are your current data governance assumptions and drivers?
- Does your governance strategy work well when your data are in the cloud?
- Do your devops processes work across on-prem and cloud facilities
- Do you have sufficient network capacity (and backup) for working with lots of your data at a cloud vendor?

Moving your data

Moving your data

- Typical needs: To/From a service lab; To a colleague; To a repository
- Relatively easy to move files from one server to another, especially on-campus
- Harder to move very large files or a large number of files, especially cross-country or internationally
- Campus bandwidths are 1 to 10 Gbps max. (125 – 1250 MB/s)
- Intercampus can be more but subject to many issues

Tools to move files and folders

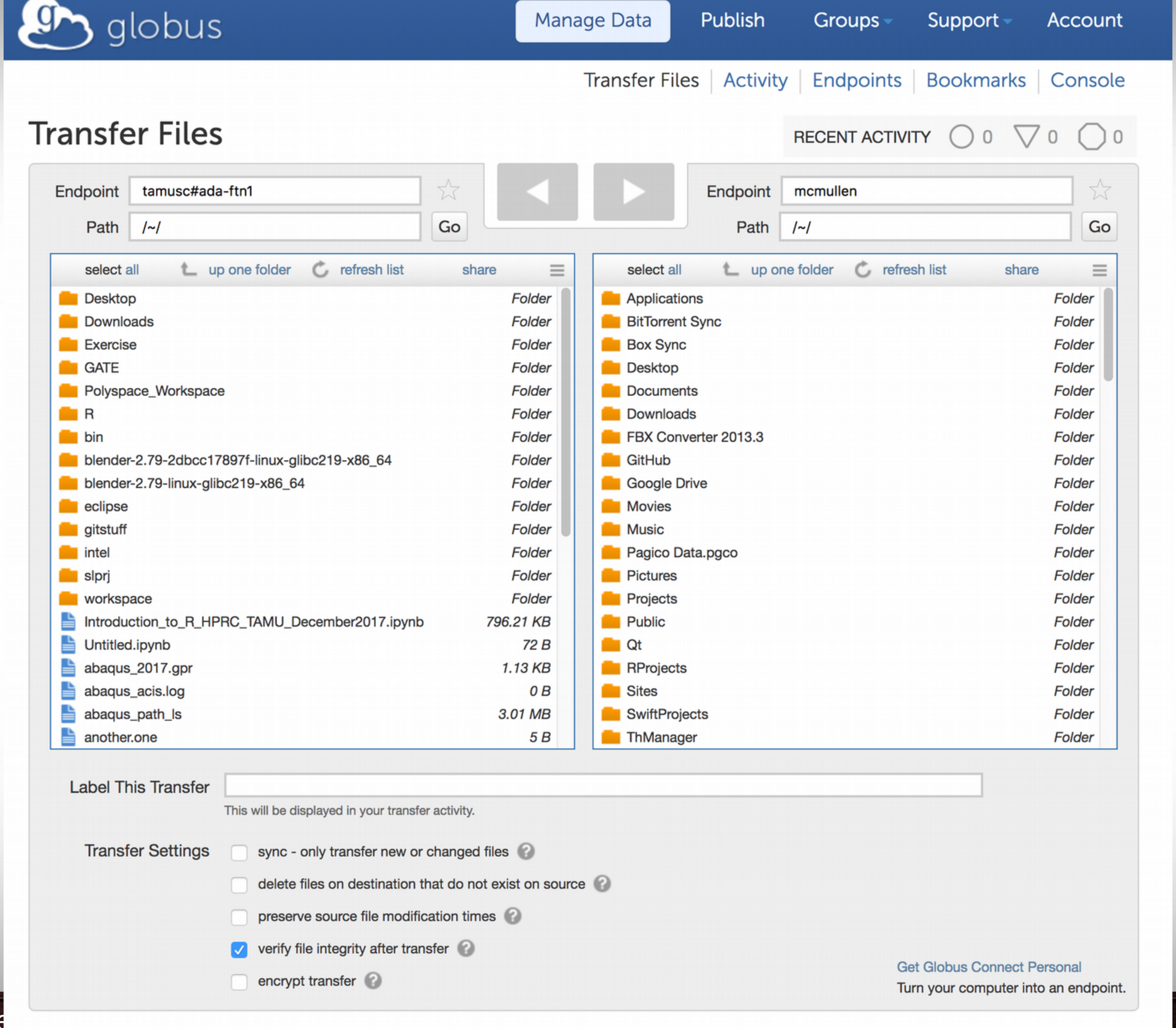
- On the same machine:
  - `mv, cp —R`
- Between machines where you have log-ins:
  - `scp, sftp, rsync, Globus`
  - tar or zip first to reduce copy time
  - Filezilla is a good GUI for moving files
  - rsync can be used to maintain up to date copies of directories and files
- From public sources
  - `wget, git`

Globus for moving files

- Globus is a service for highly optimized, reliable, and unattended file transfers
- Files are moved between “endpoints” set up by anyone
- File transfers are set up through a web interface
- You must have a globus account (free) and know what endpoints you want to use
- You can set an endpoint up on your own PC/laptop using “globus connect personal”
- Get started at globus.org

Publicly visible TAMU Endpoints

tamusc#terra-ftn	owner: tamusc@globusid.org
tamusc#ada-ftn1	owner: tamusc@globusid.org
tamusc#ada-ftn2	owner: tamusc@globusid.org
tamu#brazos-dev	owner: tamu@globusid.org
tamu#brazos	owner: tamu@globusid.org
TAMU terra-ftn	owner: tmarkhuang@globusid.org
TAMU ada-ftn2	owner: tmarkhuang@globusid.org
TAMU ada-ftn1	owner: tmarkhuang@globusid.org



Transfer Files

RECENT ACTIVITY ○ 0 ▽ 0 ⬡ 0

Endpoint: tamusc#ada-ftn1
Path: /~/ [Go]

Endpoint: mcmullen
Path: /~/ [Go]

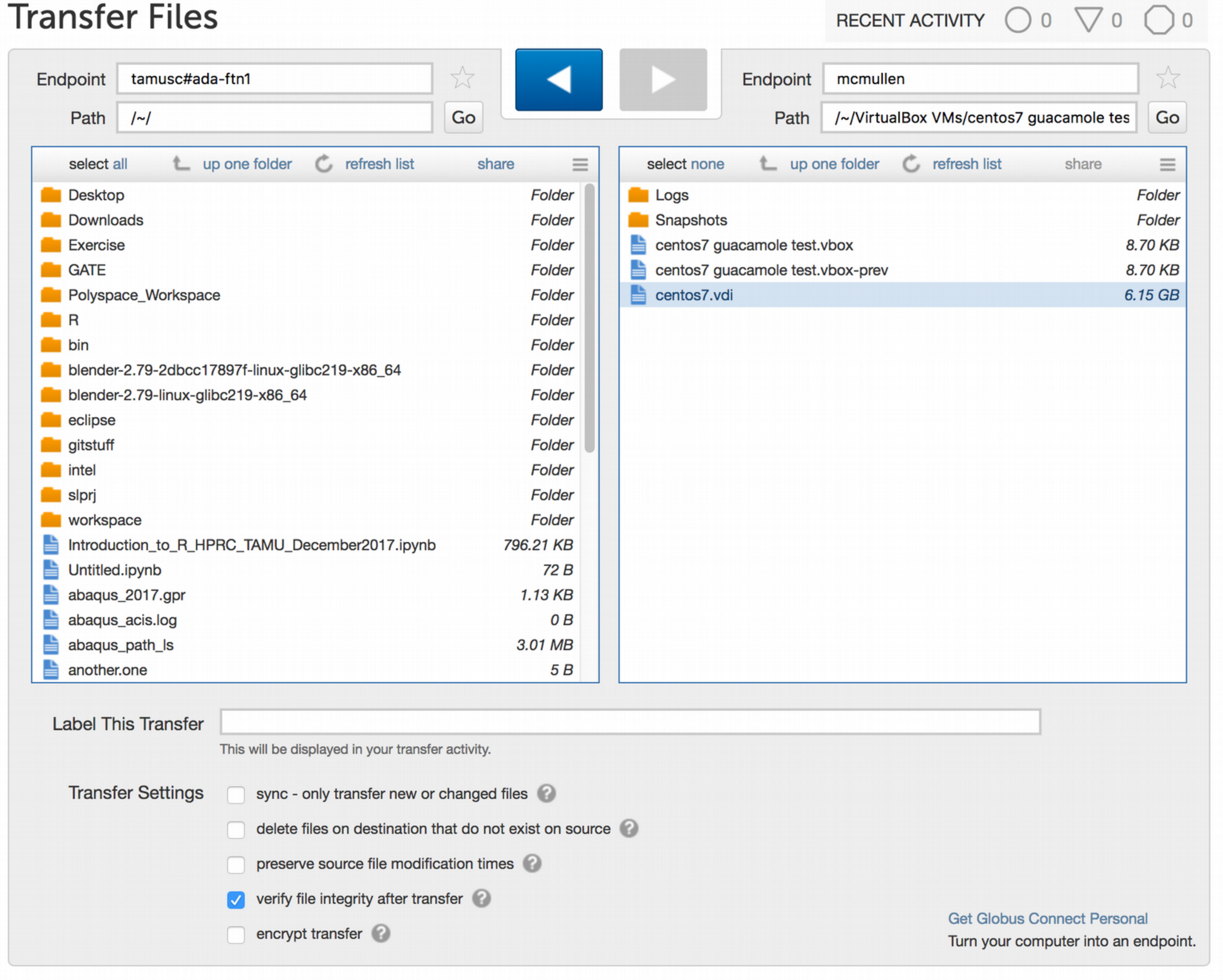
Desktop	Folder
Downloads	Folder
Exercise	Folder
GATE	Folder
Polyspace_Workspace	Folder
R	Folder
bin	Folder
blender-2.79-2dbcc17897f-linux-glibc219-x86_64	Folder
blender-2.79-linux-glibc219-x86_64	Folder
eclipse	Folder
gitstuff	Folder
intel	Folder
slprj	Folder
workspace	Folder
Introduction_to_R_HPRC_TAMU_December2017.ipynb	796.21 KB
Untitled.ipynb	72 B
abaqus_2017.gpr	1.13 KB
abaqus_acis.log	0 B
abaqus_path_ls	3.01 MB
another.one	5 B

Applications	Folder
BitTorrent Sync	Folder
Box Sync	Folder
Desktop	Folder
Documents	Folder
Downloads	Folder
FBX Converter 2013.3	Folder
GitHub	Folder
Google Drive	Folder
Movies	Folder
Music	Folder
Pagico Data.pgco	Folder
Pictures	Folder
Projects	Folder
Public	Folder
Qt	Folder
RProjects	Folder
Sites	Folder
SwiftProjects	Folder
ThManager	Folder

Label This Transfer:
This will be displayed in your transfer activity.

Transfer Settings
- ☐ sync - only transfer new or changed files
- ☐ delete files on destination that do not exist on source
- ☐ preserve source file modification times
- ☑ verify file integrity after transfer
- ☐ encrypt transfer

Get Globus Connect Personal
Turn your computer into an endpoint.



Transfer Files

RECENT ACTIVITY ○ 0 ▽ 0 ⬡ 0

Endpoint: tamusc#ada-ftn1 ☆
Path: /~/ Go

◀ ▶

Endpoint: mcmullen ☆
Path: /~/VirtualBox VMs/centos7 guacamole tes Go

select all | up one folder | refresh list | share

Desktop	Folder
Downloads	Folder
Exercise	Folder
GATE	Folder
Polyspace_Workspace	Folder
R	Folder
bin	Folder
blender-2.79-2dbcc17897f-linux-glibc219-x86_64	Folder
blender-2.79-linux-glibc219-x86_64	Folder
eclipse	Folder
gitstuff	Folder
intel	Folder
slprj	Folder
workspace	Folder
Introduction_to_R_HPRC_TAMU_December2017.ipynb	796.21 KB
Untitled.ipynb	72 B
abaqus_2017.gpr	1.13 KB
abaqus_acis.log	0 B
abaqus_path_ls	3.01 MB
another.one	5 B

select none | up one folder | refresh list | share

Logs	Folder
Snapshots	Folder
centos7 guacamole test.vbox	8.70 KB
centos7 guacamole test.vbox-prev	8.70 KB
centos7.vdi	6.15 GB

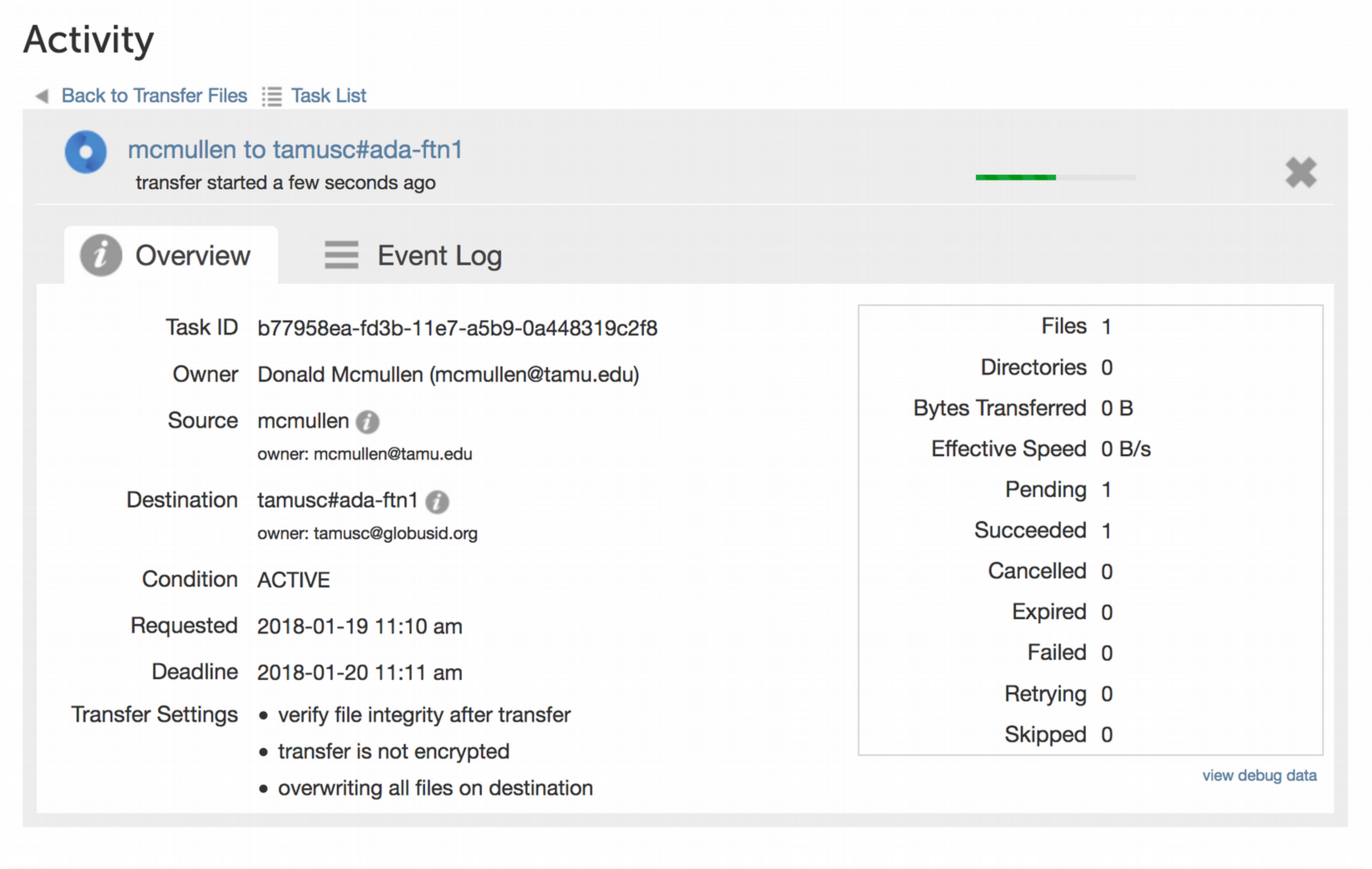
Label This Transfer

This will be displayed in your transfer activity.

Transfer Settings

- □ sync - only transfer new or changed files ?
- □ delete files on destination that do not exist on source ?
- □ preserve source file modification times ?
- ☑ verify file integrity after transfer ?
- □ encrypt transfer ?

Get Globus Connect Personal
Turn your computer into an endpoint.



Activity

mcmullen to tamusc#ada-ftn1

transfer started a few seconds ago

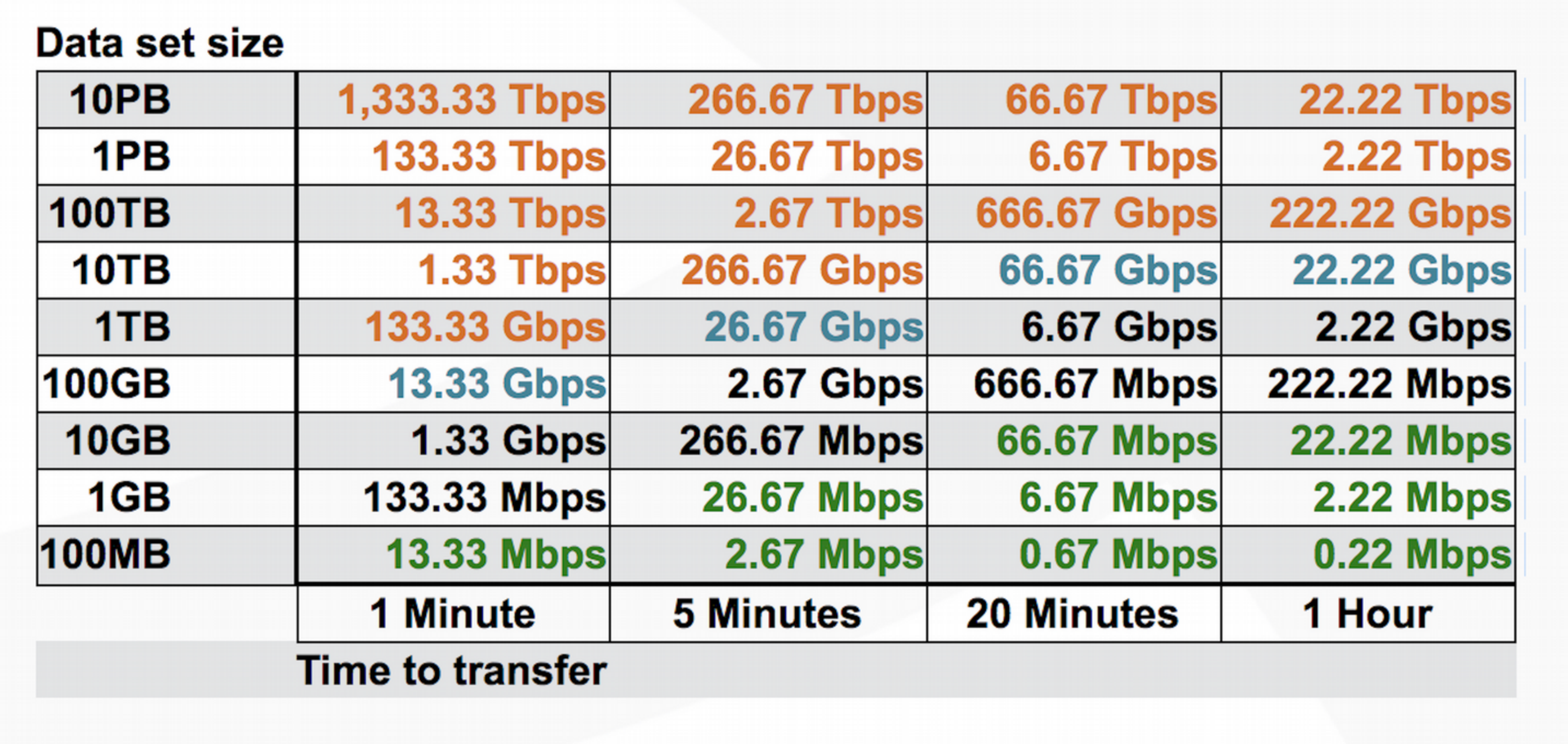
Task ID	b77958ea-fd3b-11e7-a5b9-0a448319c2f8
Owner	Donald Mcmullen (mcmullen@tamu.edu)
Source	mcmullen owner: mcmullen@tamu.edu
Destination	tamusc#ada-ftn1 owner: tamusc@globusid.org
Condition	ACTIVE
Requested	2018-01-19 11:10 am
Deadline	2018-01-20 11:11 am
Transfer Settings	• verify file integrity after transfer • transfer is not encrypted • overwriting all files on destination

Files	1
Directories	0
Bytes Transferred	0 B
Effective Speed	0 B/s
Pending	1
Succeeded	1
Cancelled	0
Expired	0
Failed	0
Retrying	0
Skipped	0

Minimum Time to Transfer Data

Minimum time needed to to transfer 1 Terabyte of data across various speed networks:

10 Mbps network	**300 hrs (12.5 days)**
100 Mbps network	**30 hrs**
1 Gbps network	**3 hrs**
10 Gbps network	**20 minutes**



Data set size

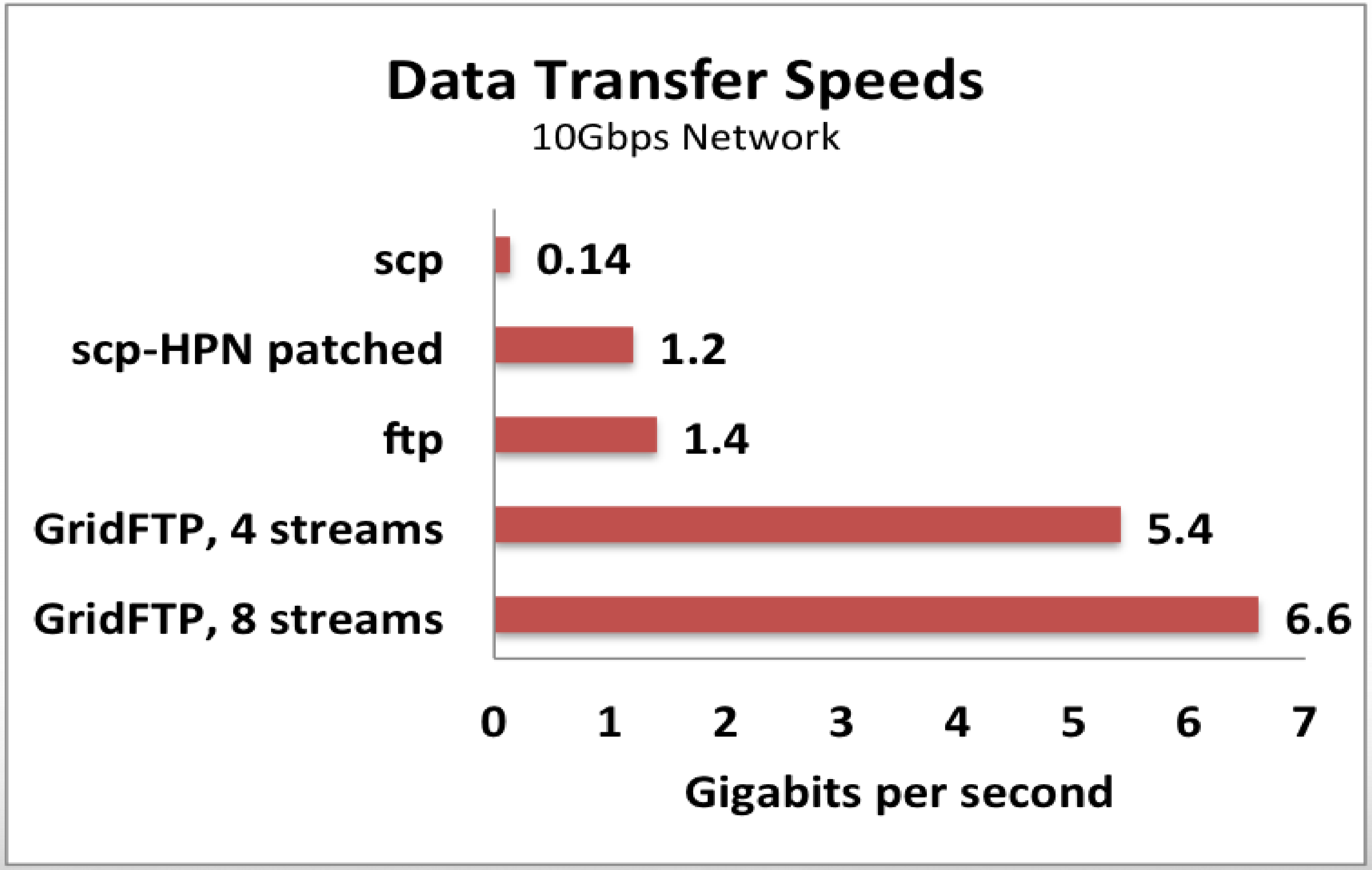
Data set size	1 Minute	5 Minutes	20 Minutes	1 Hour
10PB	1,333.33 Tbps	266.67 Tbps	66.67 Tbps	22.22 Tbps
1PB	133.33 Tbps	26.67 Tbps	6.67 Tbps	2.22 Tbps
100TB	13.33 Tbps	2.67 Tbps	666.67 Gbps	222.22 Gbps
10TB	1.33 Tbps	266.67 Gbps	66.67 Gbps	22.22 Gbps
1TB	133.33 Gbps	26.67 Gbps	6.67 Gbps	2.22 Gbps
100GB	13.33 Gbps	2.67 Gbps	666.67 Mbps	222.22 Mbps
10GB	1.33 Gbps	266.67 Mbps	66.67 Mbps	22.22 Mbps
1GB	133.33 Mbps	26.67 Mbps	6.67 Mbps	2.22 Mbps
100MB	13.33 Mbps	2.67 Mbps	0.67 Mbps	0.22 Mbps

Time to transfer

Source: http://fasterdata.es.net/home/requirements-and-expectations

Use the right tool…

Berkeley, CA ←→ Argonne, IL   RTT=53



Some lessons learned and observations about storage systems

- Regardless of where you work, all those file types will follow
  - Small files
  - Large files (>600GB)
  - Directories with millions of files
  - Spreadsheets
  - “Structured” flat files
  - Very large binary files
  - Very large text files
- But a given storage system will usually only handle a few of these well
- OK, then what about Metadata? Keeping track of your stuff will need attention, thought, planning and automation. “Storage is cheap, Metadata are precious.” (The next thing, may be big.)

Source: Riffing on Chris Dadigian, Bioteam.

More observations

- Shipping disk drives is dangerous for your data, though Amazon will come and pick it up for you (https://aws.amazon.com/snowmobile/).
  Great bandwidth, terrible latency.

- Using the network (under the right conditions) is still the better option.
- There is no easy way to determine a file's "goodness" except hashes or checksums, although these can be automated to an extent, e.g. Globus (globus.org), during network copies.

Summary

- Who knew data management was so complicated?
- In research, data management is critical to success, or lack of attention can lead to trouble
- Three main aspects of data management are
  - How/where you store your data
  - How you annotate your data for understanding and findability
  - Moving your data has some non-trivial aspects if you have a lot of it
- An emerging part of data management is saving computational methods and code; can be done now by saving fully-configured virtual machines.

Questions?