

Introduction to High Performance Computing (HPC) – Session 1

using the "**Computational Shared Facility**" (CSF)

Course materials / slides available from:

<https://ri.itservices.manchester.ac.uk/course/rcsf/>

Research Platforms, IT Services

its-ri-team@manchester.ac.uk

<https://ri.itservices.manchester.ac.uk/csf3/>

Introduction to High Performance Computing (HPC) – Session 1

using the "**Computational Shared Facility**" (CSF)

Course materials / slides available from:

<https://ri.itservices.manchester.ac.uk/course/rcsf/>

Research Platforms, IT Services

its-ri-team@manchester.ac.uk

<https://ri.itservices.manchester.ac.uk/csf3/>

Housekeeping

- Please let me know if you're leaving
 - 10am - 12:30pm (practical sessions 1, 2, & 3)
 - Lunch *approx* 12:30pm - 1:30pm
 - 1:30pm - 4pm (practical sessions 4 & 5)
- 1-to-1 help is available if needed during exercises.
- Power adapters are available for the purposes of charging laptops, please be considerate of other users and *be careful of any trailing leads*.
- Got a question at any point? PLEASE ASK!!

Feedback

- Your feedback is important to us!
- Please give feedback on this course
 - Quick form at <https://goo.gl/forms/zfZyTLw4DDaySnCF3>
(choose "*Introduction to HPC (Using CSF)*")
 - Feedback is important to help us improve our courses
 - Records your attendance on the course

Who we are - Research IT

<https://research-it.manchester.ac.uk/services/>

MANCHESTER
1824
The University of Manchester

Research IT

Search

News Services Communities Events Acknowledgement Contact Tags

Research IT Services

Research IT offers a wide range of services to researchers and PGRs at the University. Some of these services incur costs which should be included in research grant applications after a consultation with us. Some services are free of charge or free to a certain limit. Please check the service that you wish to use and get in touch with us if you would like to discuss your requirements further.

The Computational Shared Facility (CSF)

Provides a batch-based and limited interactive on-campus platform with a large range of software for all research domains to enable computationally-intensive work. Integrated with Research Data Storage (RDS).

[Find out more](#)

Research Software Engineering

The Research Software Engineering (RSE) team enhances the University's capacity to produce high quality research software by collaborating with researchers to create correct, efficient, readable, reliable and sustainable code.

[Find out more](#)

Application Support

Provides help with installing, licensing, building, or running of research applications. This includes both commercial applications used in research and open source, or research written applications.

[Find out more](#)

The Interactive Computation Shared Facility (ICSF)

Provides an interactive on-campus platform with a large range of software for all research domains to enable computationally-intensive work. Integrated with Research Data Storage (RDS).

[Find out more](#)

The HPC Pool

Provides a batch-based on-campus platform with a large range of software for all research domains to enable computationally-intensive work. Integrated with Research Data Storage (RDS).

[Find out more](#)

Research Data Storage

Centrally hosted, resilient, backed-up, data storage for research staff and PGR students.

[Find out more](#)

Who the course is for - everyone

- Those new to HPC / research computing / who want to try the CSF out?
 - We'll introduce you to these topics and you'll try it out today
- Maybe your supervisor asked you to get a CSF account
 - We'll teach you how to use it
- Those who have used CSF already but want to know more?
 - Parallel jobs, job arrays, the batch system, ...
- *CSF usage is today's focus so you can use it effectively in your work*
 - *not* theoretical aspects of HPC
 - but we'll explain some of the basics to help you make good use of the resources
 - *not* parallel software development or version control
 - But we'll show you how to run high-end parallel applications
 - *not* Linux installation / administration
 - but we'll cover the basic Linux commands needed to use the CSF
 - *not* the specifics of the software you plan on using
 - ask your PI/supervisor for help with that

Invest a little time now, get results much faster!

WHY & WHAT ...

High Performance Computing : why use it & what is it

Motivation: Why use HPC (and the CSF)?

- Some (most?) research computation not suitable for your desktop/laptop
 - Takes *too long* to run
 - Needs more memory
 - Uses *too much* disk/storage space
- Use advanced centrally-run UoM hardware
- Eventually use regional / national supercomputers

Do not let the size/capacity/power of your computer dictate the size and complexity of the models/simulations/systems/problems you are solving!

What is HPC?

- insideHPC.com
 - *High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.*
- HPC systems are usually a cluster of compute nodes (with some extra items such as login nodes, storage, networking)
 - The CSF matches this description!

A new way of working!



Running on a desktop

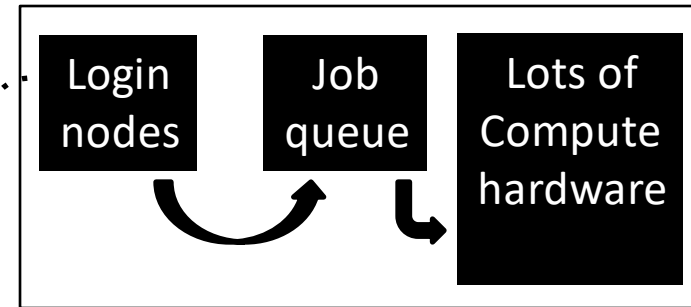
- You can fire up a GUI, run an app immediately. BUT:
- Got enough memory, cores, storage?
- Need to keep the PC to yourself (public cluster PC?)
- For several days?!
- Only one "job" (simulation, analysis) at a time?



vs

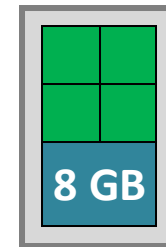
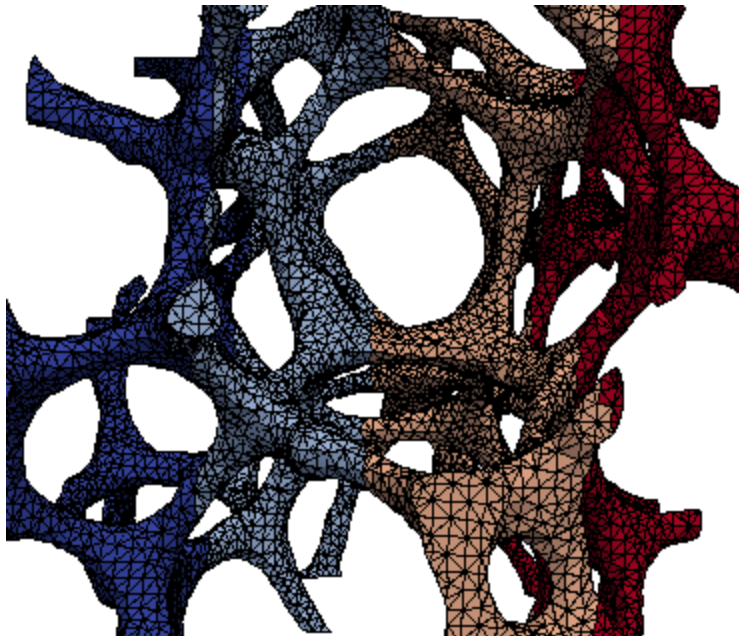
Logging in to the HPC system

- Submit "**jobs**" to the queue
- Jobs wait until selected to be run
- Jobs run on *high-end* hardware (lots of cores, memory, disk)
- Jobs run safely for days
- Many jobs can run at once
- Can log out any time (jobs still run)
- Log in to check on progress, get results

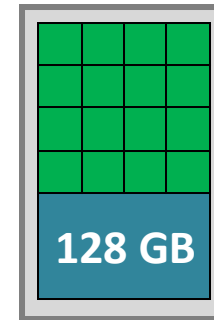


Job Example: Finite Element Analysis

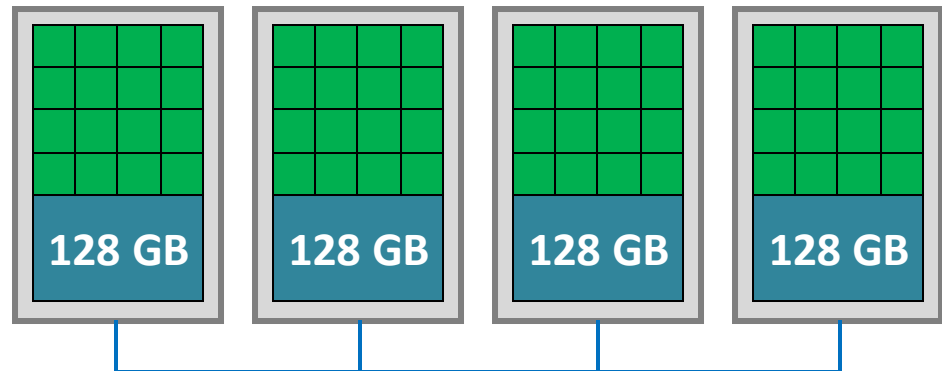
- Perform stress analysis on 3D mesh
 - App s/w splits mesh into chunks
 - It performs calculations on chunks, in *parallel*
 - Faster and/or larger problem size



Desktop: 4 cores,
8GB RAM.
8 days to complete



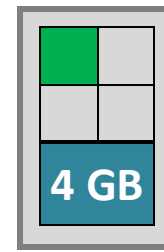
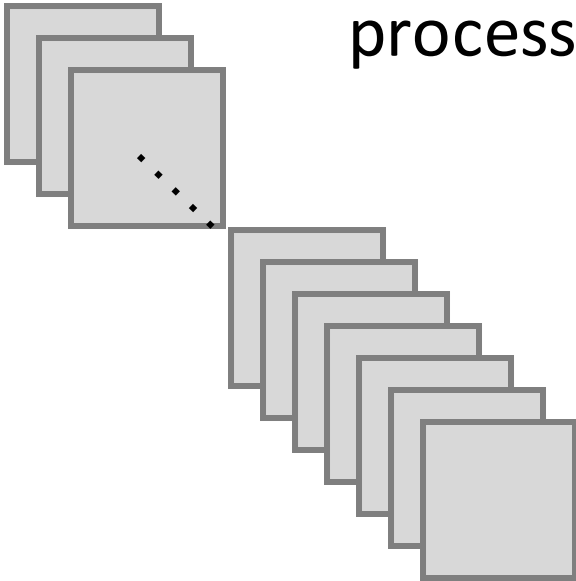
Single HPC compute
node: 16 cores,
128GB RAM
~2 days to complete



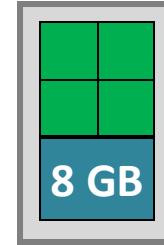
Multiple HPC compute nodes:
64 cores, 512GB RAM
~0.5 days to complete

Job Example: Image Analysis

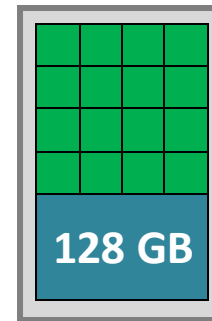
- High Throughput Computing
 - Not all apps do "HPC" / parallel
 - But you have *lots* of data
 - Each image takes 1hr to process (and are independent - can be processed in any order)



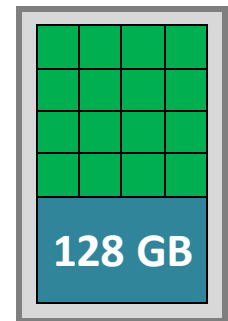
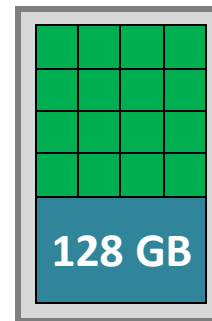
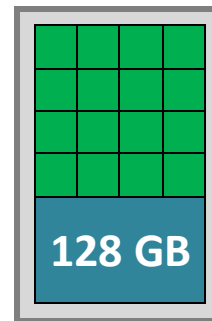
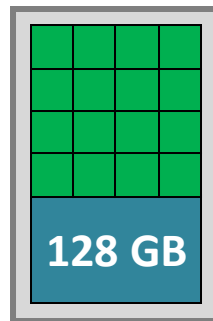
Laptop: 1 copy of software running on 4-core CPU.
Over 1 year to complete!!



Desktop: 4 cores, 4 copies of software running.
~100 days to complete!



Single HPC compute node: 16 copies of software running.
~26 days to complete



Multiple HPC compute nodes:
64 copies of software running.
~6 days to complete

Example: 10,000 image scans to be analysed by an image processing application. Each image takes *1 hour* to process.

What we'll be using today - the CSF

- Q: who has used the Computational Shared Facility (CSF) before?
- Currently: CSF3
 - A large Linux cluster system
 - 8,644 CPU cores (all Intel Xeon CPUs)
 - 100 GPUS (68 Nvidia v100s + 32 Nvidia A100s)
 - Got big datasets to process? Can run large-memory jobs
 - (we'll cover all of these details throughout the course)

BUT, you do not have to be running huge parallel jobs, or be a Linux / HPC expert, to use our systems and to benefit from "HPC" and the CSF

Who can use the CSF?

The following info is mainly for people who may want to "buy in" to the CSF. Your PI/supervisor or School may well have already done this! **If interested, ask us at the end of the course.**

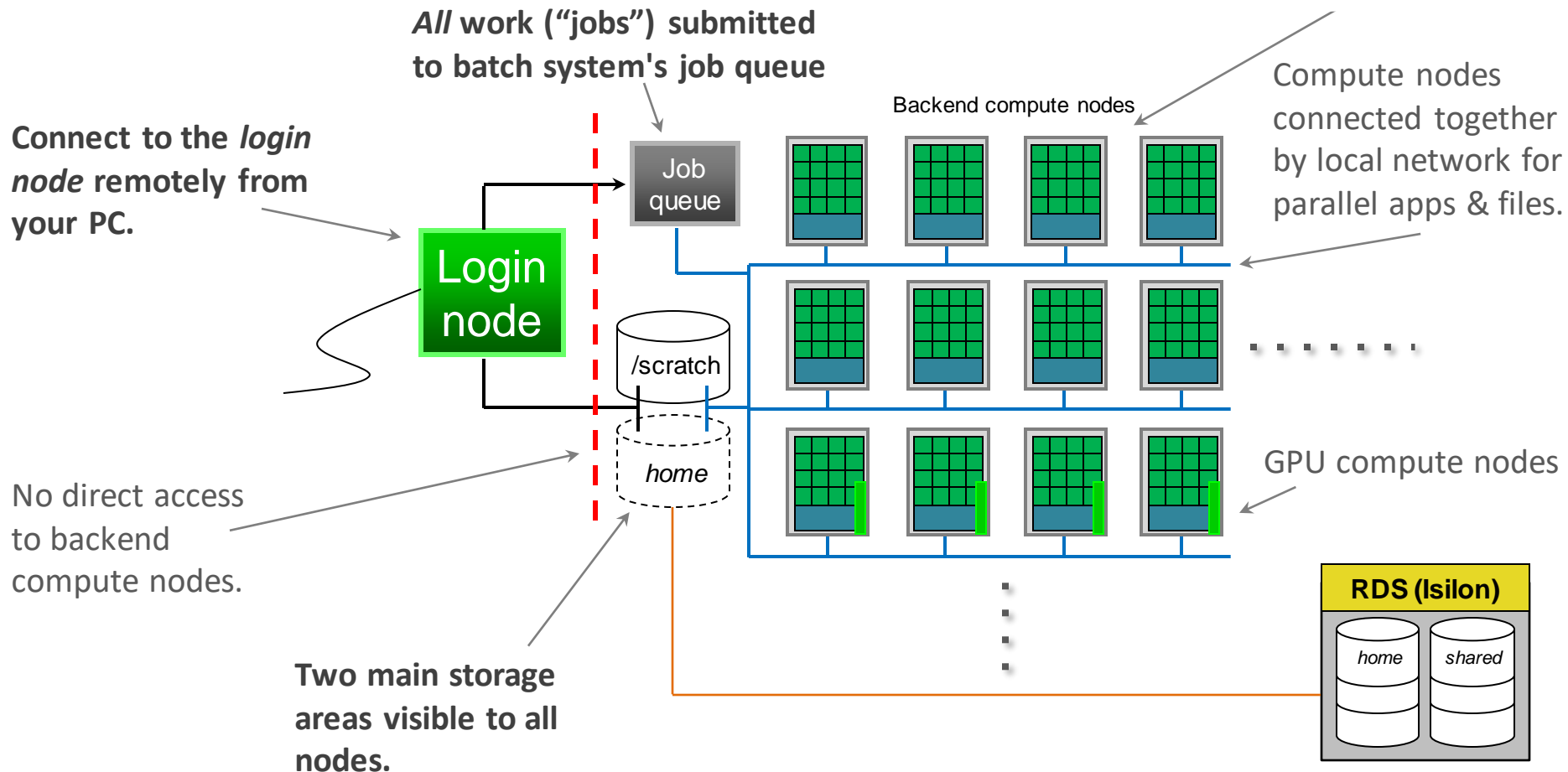
- CSF uses a *shared funding model*
 - Researchers/academics/schools *contribute financially* to buy compute hardware
 - All h/w pooled so that all users can access the h/w
 - H/w *not* associated with individuals so it can always be in use as long as there are jobs to run!
- The time it takes to run all of your jobs depends on the size of the contribution with which you are associated
 - A research group that contributes more will be able to get more jobs done *sooner*
 - *Managed automatically by the batch system – you just submit jobs!*
- Some limited 'free at the point of use' access for non-contributors

CSF: THE BASICS...

Hardware, OS, logging in, security, home filesystem, copying files, Linux, GUIs

What is the CSF?(more details)

- **Computational Shared Facility**
- A *batch* compute *cluster* to run your "jobs" (simulations, analysis,...)
- Here are the main components you'll learn about: 100s of compute nodes



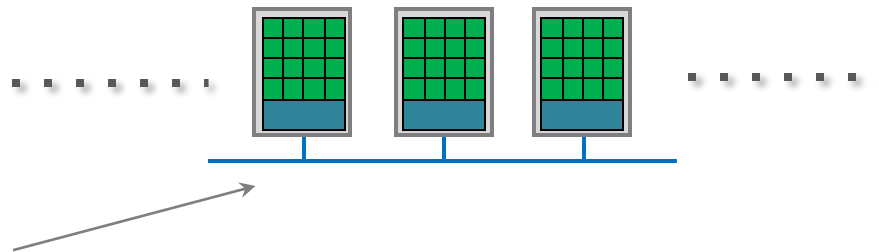
Compute Nodes

- The CSF is a cluster of 100s of "*compute nodes*"

Think of a compute node as **high end PC** (the CSF provides 100s of these!):

- Multi-core CPUs (e.g., 16, 28, 32 cores)
- Lots of RAM (e.g., 128GB, ..., 2TB!)
- Network (possibly fast *InfiniBand* n/w)
- OS (Linux)
- Local disk (for temp files)
- Maybe a GPU

Not all nodes have the same h/w. But the CSF hides these details – let it choose which nodes you use, or you can specify certain requirements.



Local network connecting compute nodes. This allows:

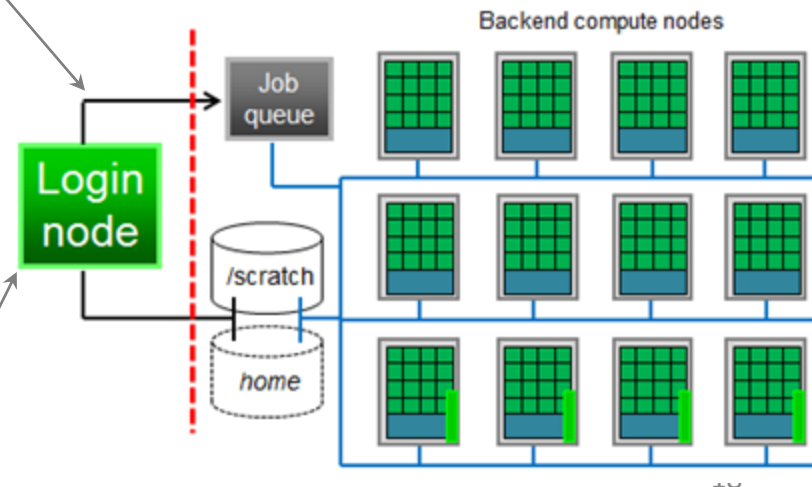
- Some applications can use more than one node! (e.g., for big data / simulations)
- All compute nodes see the same files
- **You don't need to transfer files to a compute node!**

Login Nodes

- CSF: More than 400 *backend* compute nodes
 - Too many for *you* to find a free compute node to use
 - Simpler: connect to a login node, let *job-queue* find a node

Key concept!

- **Do:** Submit work ("*jobs*") to be run
 - No direct access to compute nodes
 - Submit jobs to the *job queue*.
 - The system will run your work on available *compute nodes* meeting your requirements.
- **Do not:** run applications on the login node
 - Shared by all users, not much memory
 - For *lightweight* tasks (job submission, file transfer, ...)



Some pictures of the CSF

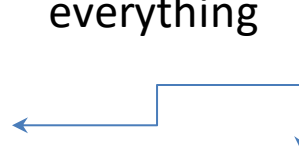


↑
Compute nodes

Local storage →



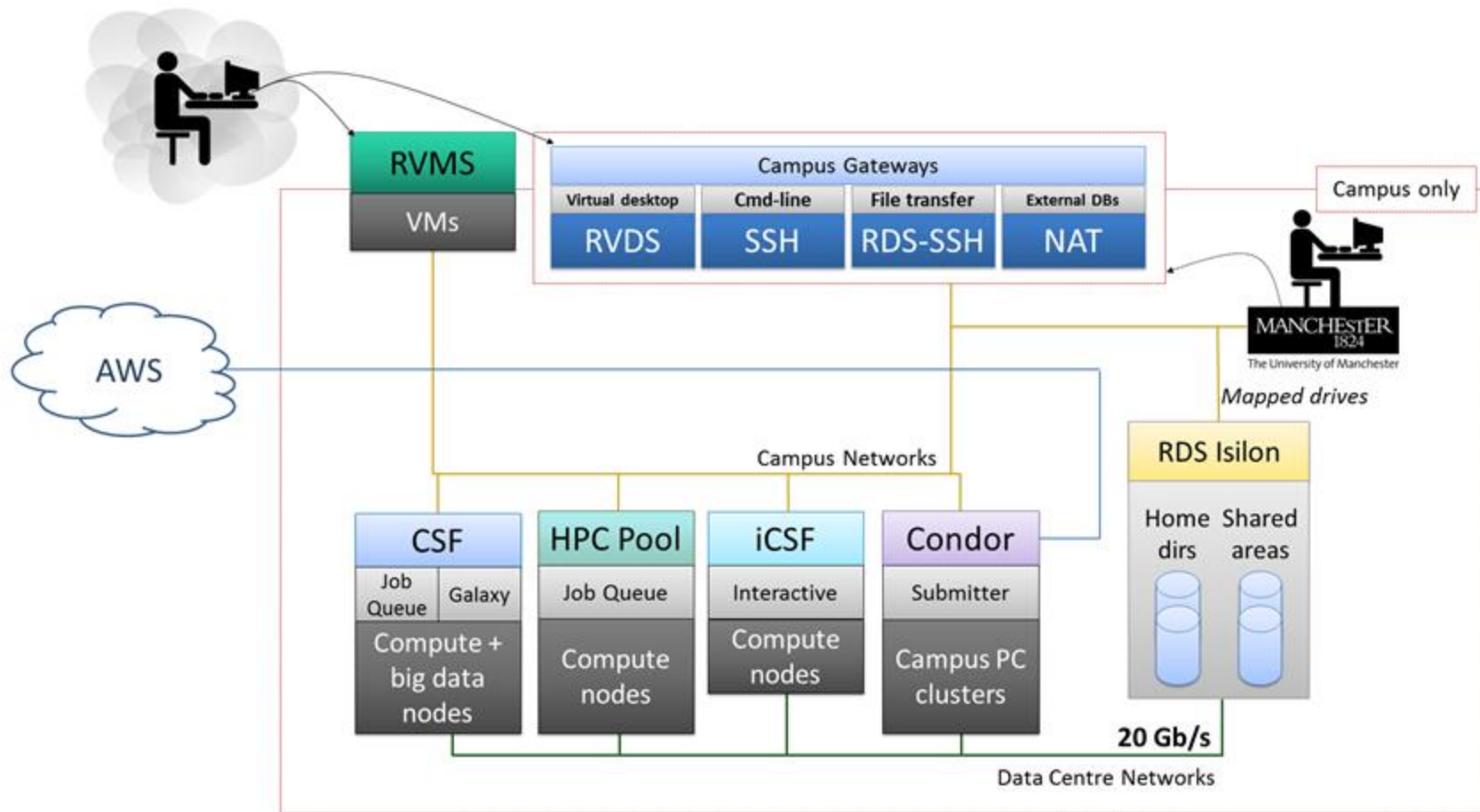
Networking connects everything



Features of the CSF

- Variety of software applications (120+) & compilers
 - Gromacs, MATLAB, Abaqus, Bowtie2, Gaussian, Fluent, many more
- 1-core (serial) jobs or *many*-core (parallel) jobs
 - Do more work. Get it done sooner (see later!)
- Lots of RAM: 64GB, 128GB, 192GB, 256GB, 512GB, 1TB, 1.5TB, 2TB, 4TB (!!) compute nodes
- Backed-up file storage (no more USB disks!!!!!!!!!!!!)
- Hardware failures don't stop your research
- Leave computational work running (for days)
- 8,644 cores currently in the system
- Dedicated support team

Part of the *Computationally Intensive Research Ecosystem*

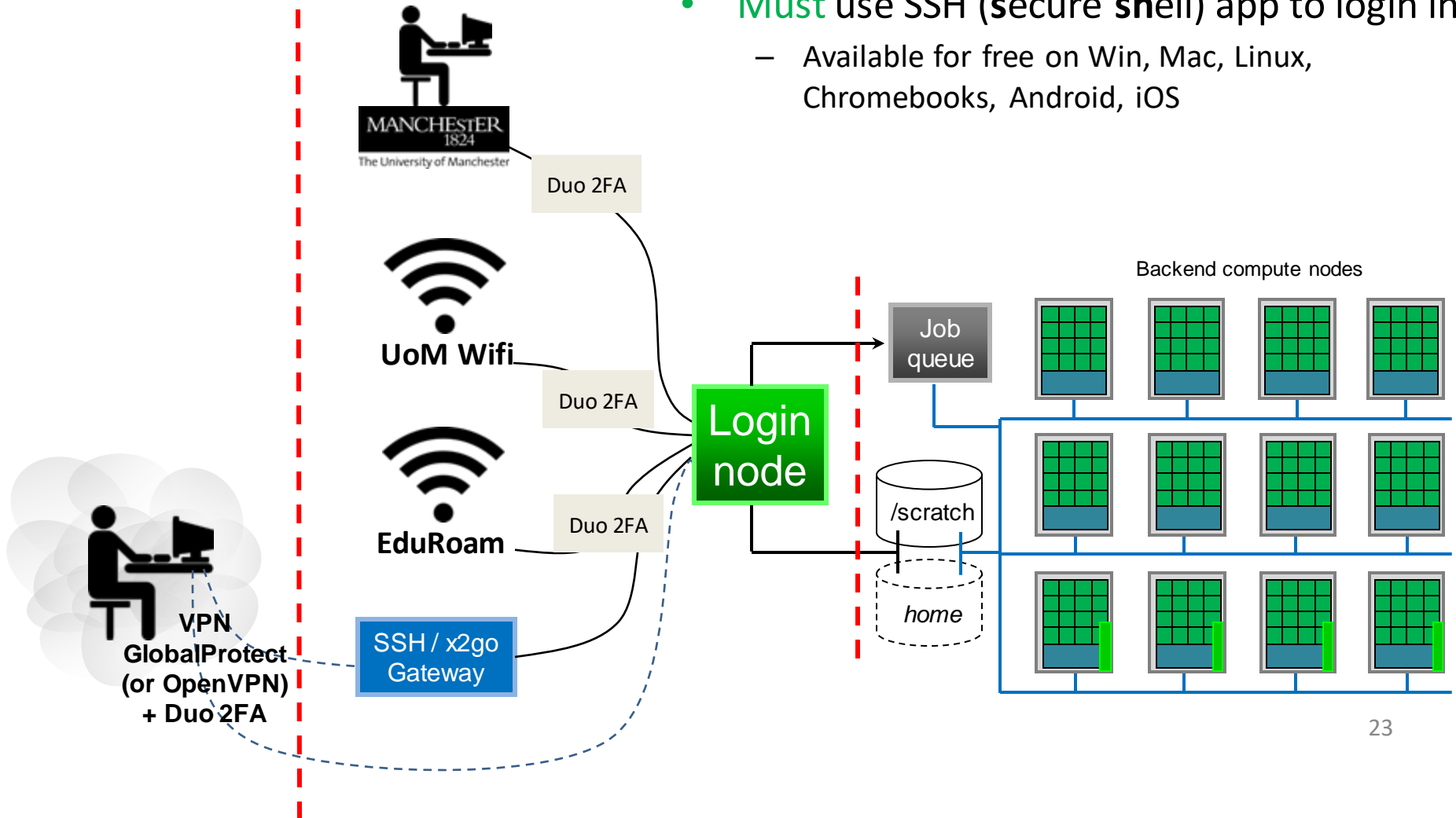


Logging in ...

Let's get access to the system

Where can I log in from?

- **Must** use SSH (secure **shell**) app to login in
 - Available for free on Win, Mac, Linux, Chromebooks, Android, iOS



Security (1)

- The CSF has a private (campus only) IP address
 - Firewall also controls connections to and from the system
- When you are on campus
 - Connect from any PC/laptop with a wired connection, or University WiFi, or Eduroam WiFi
 - Does not matter if using VPN or not, but you will always be asked to authenticate using your 2FA (DUO) device, e.g.

```
keyring@i-ulit636838l ~ $ ssh -X keyring@csf.itservices.manchester.ac.uk
Password:
Duo two-factor login for keyring@

Enter a passcode or select one of the following options:

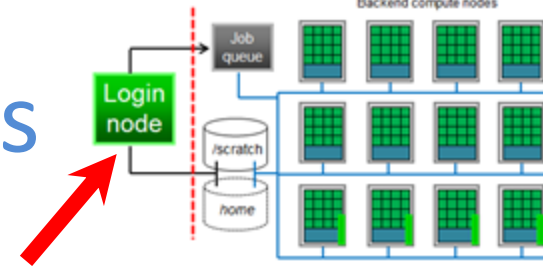
1. Duo Push to +XX XXXX XX6353

Passcode or option (1-1): 1
```

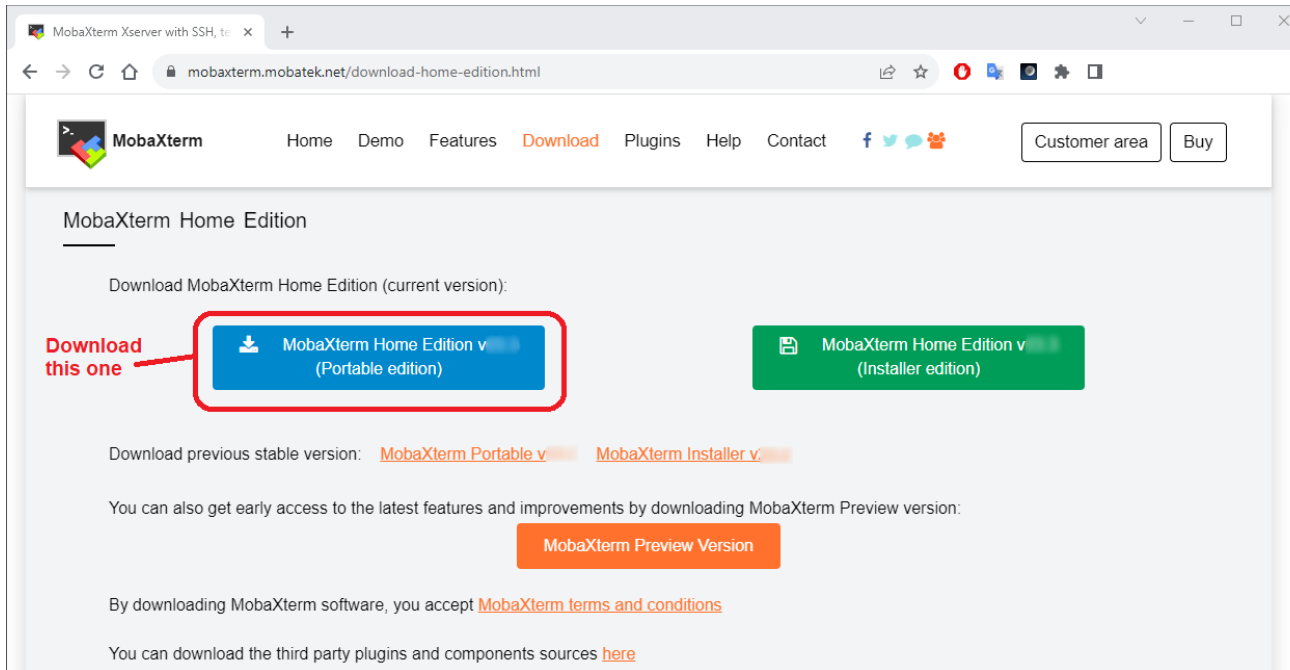
- When you are off campus
 - May not be available to all UoM account types *today*
 - Sign-in to University Global Protect VPN + DUO 2FA
 - Allows you to make a direct connection to CSF
- Further documentation:

<https://ri.itservices.manchester.ac.uk/csf3/getting-started/connecting/>

Connect to CSF from Windows



- Access the CSF from a PC / laptop using an SSH (Secure **Shell**) app
 - Sometimes called a "terminal".
 - There's no web-site or other fancy GUI on the CSF – use the "command-line".
- **Windows users** need to install a free *terminal* app called MobaXterm
- <https://mobaxterm.mobatek.net/download-home-edition.html>
the **Home edition** (**portable edition**) does *not require* Administrator rights - just *extract* the small .zip file in your P-Drive or USB stick for example.



1. Download using the **blue** box.
2. Once downloaded, *right-click* on the .zip file and select:

"Extract all ..."

This will *unpack* the .zip file to a folder.

MobaXterm

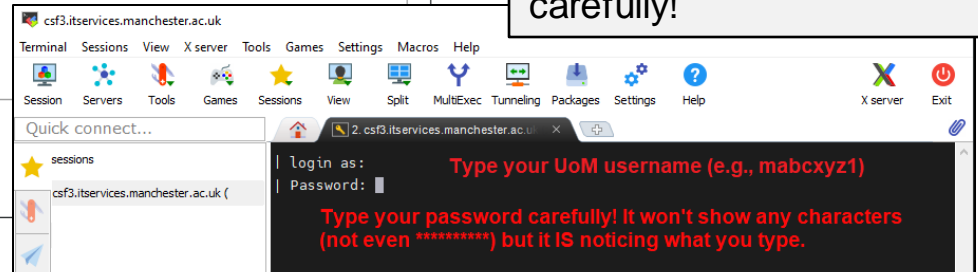
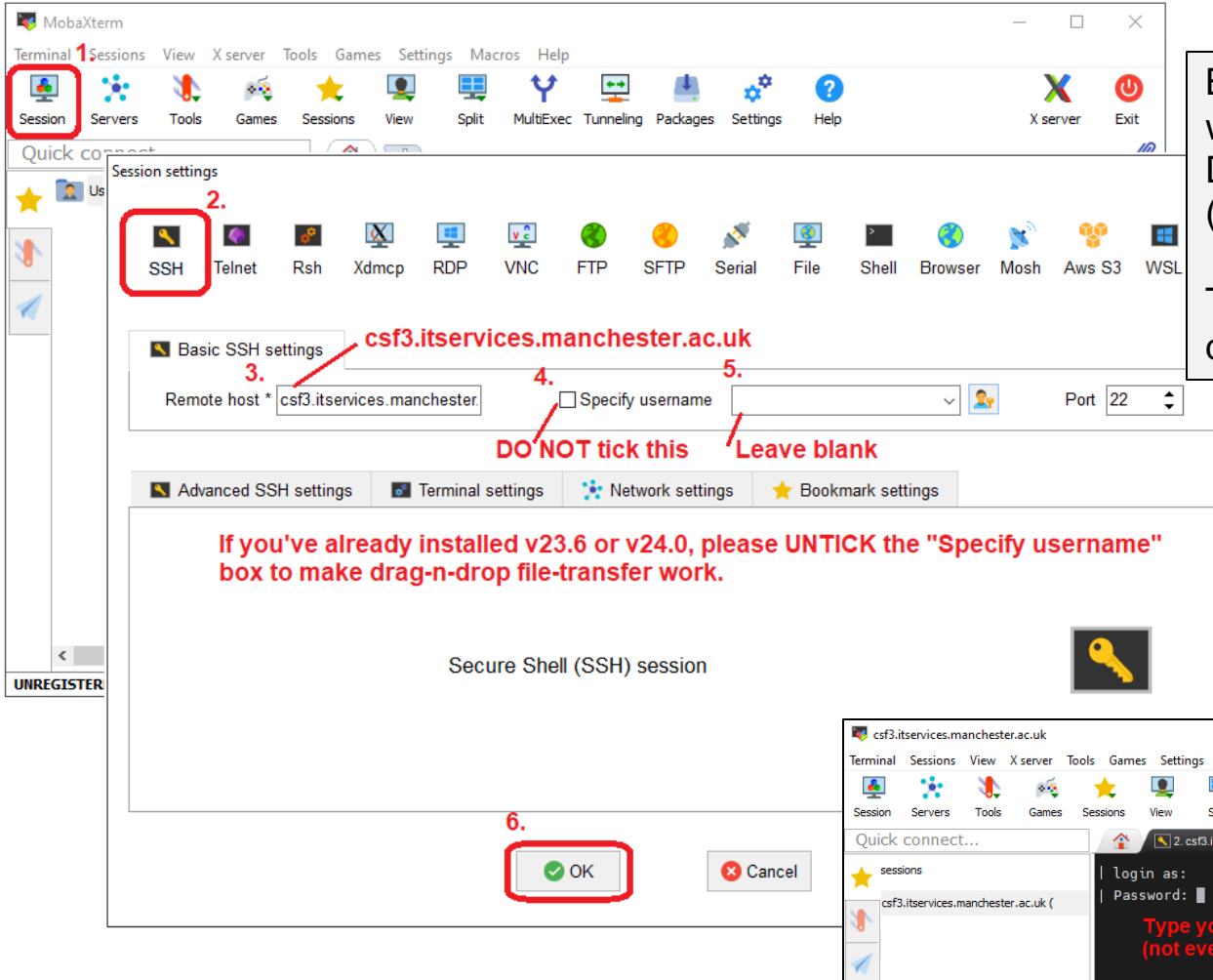
A. After **extracting** the .zip file, start MobaXterm_Personal_xy.z (double-click on the icon)

B (1-6). Create a "Session" which saves the CSF's details. **DO NOT** save your username (due to a bug in MobaXterm.)

This is needed to make file drag-n-drop work (see later.)

C. This will then start to log you into the CSF – it will ask for your usernames and password. Type carefully!

D. See slide about 2FA – you may be asked for DUO after your password

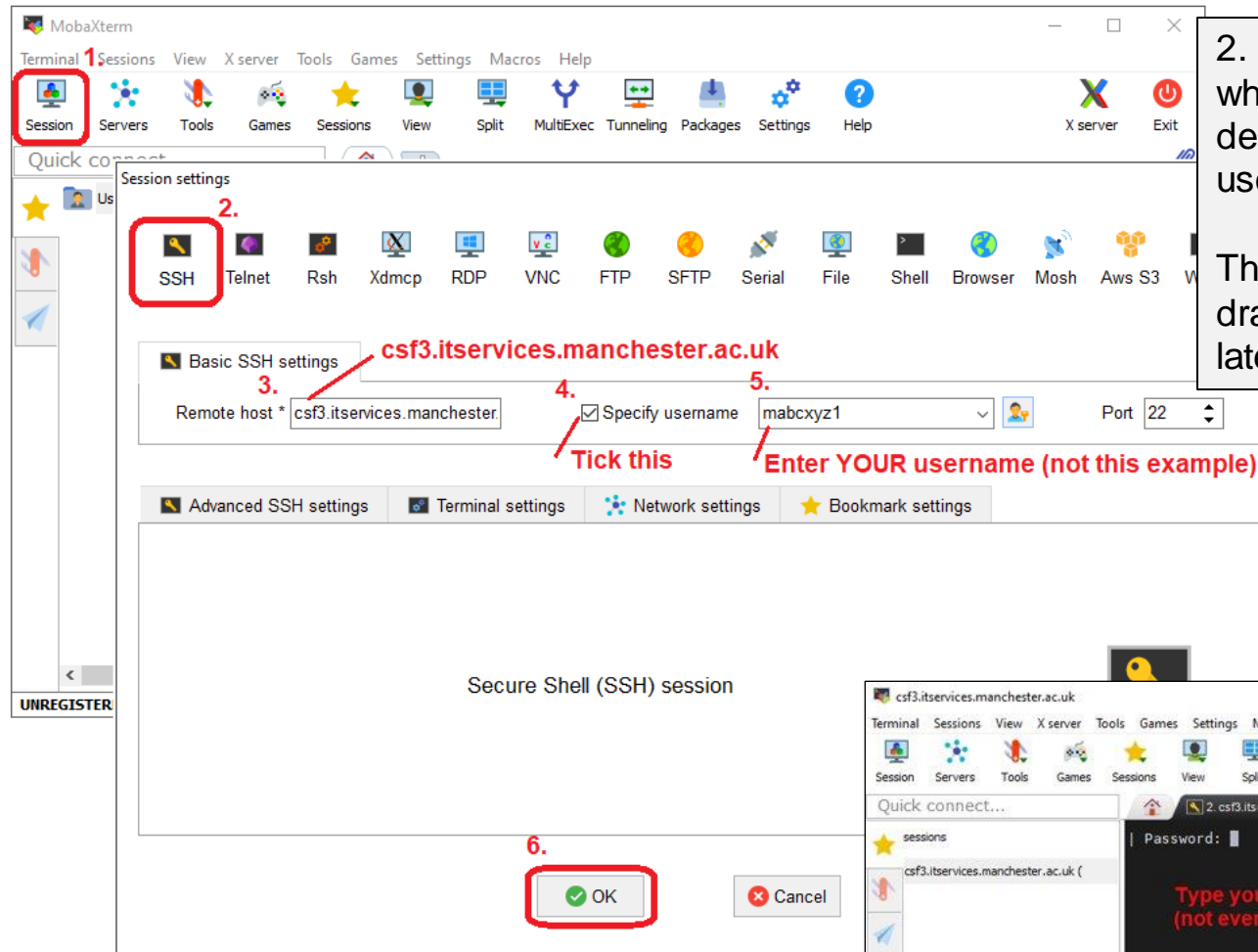


MobaXterm

1. After extracting the .zip file, start MobaXterm_Personal_xy.z (double-click on the icon)

2. Create a "Session" which saves the CSF's details along with your username.

This is needed to make file drag-n-drop work (see later.)



3. This will then start to log you into the CSF – it will ask for your password. Type carefully!

Type your password carefully! It won't show any characters (not even *****) but it IS noticing what you type.

4. See slide about 2FA – you may be asked for DUO after your password

Do you want to save password for [redacted]@csf3.itsservices.manchester.ac.uk?



Yes

No

If you want maximum security for your stored password, you can define a "master password" by going to ["Settings" -> "Misc" tab -> "MobaXterm passwords settings"](#)

☒ Do not show this message again

If asked to save your password, we recommend you say "No", for security.

Drag-n-drop file browser for upload / download

(new users won't have as many items in the list!)

We're on (one of) the CSF login nodes. Any commands you use will be typed "at the prompt", which shows your username and current directory (folder.)

csf3.itsservices.manchester.ac.uk ([redacted])

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

/mnt/users01/support/ [redacted]

Name

- ..
- .alces
- .ansible
- .ansys
- .apptainer
- .aspera
- .cache
- .cfx
- .chainer
- .checkm
- .chimera
- .cmake
- .compucell3d_py3
- .comsol
- .comsol_old_173471
- .conda
- .conda.csf3
- .conda.old
- .config
- .continuum
- .cpan
- .cpan-ignore2
- .cpanm
- .cst-workdir
- .cst2012
- .cupy
- .cytoscape
- .dart
- .dbus
- .dirm2nii

Welcome to CSF3

Docs: <https://ri.itsservices.manchester.ac.uk/csf3/getting-started>

Help: its-ri-team@manchester.ac.uk

*** REMINDER: Scratch Tidy In Operation ***

Reminder that scratch **cannot** be used for long term storage. Files not used (not read or written by you or your jobs) for 3 months or longer will be removed. However, please **note** that if you have recent jobs reading very old datasets, those datasets will NOT be deleted.

!!! You may have files at risk !!!

Use your CSF 'home' dir or RDS for keeping **important** files long term. These areas are backed up. Scratch is NOT backed up.

Jan 2023: New - check your scratch usage (space consumed and number of files) by running the following command on the login node: `scrusage`

19th June 2023: Due to the cyber-incident the University has turned off the web-proxy. Therefore, users will NOT be able to access external websites, repositories etc. via the web-proxy. If external access is required, please use a batch job or interactive session on a compute node (via qssh.) For more info on doing this please see: <https://ri.itsservices.manchester.ac.uk/csf3/batch/qssh/>

22nd Aug 2023: All access to scratch and RDS (isilon) storage has been restored and batch jobs are running normally. Please check the outputs of any recent jobs to ensure they completed as expected.

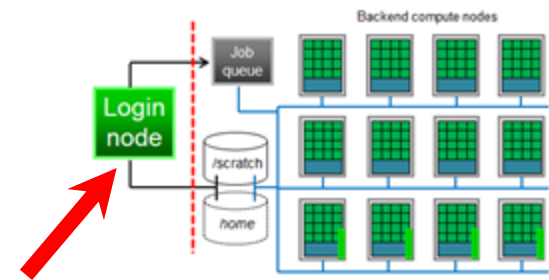
6th Sept 2023: scratch performance issues have been resolved.

Please read all notices above

[redacted]@login1 [csf3] ~]\$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Connect to CSF from Linux



- Access the CSF from a PC / laptop using an SSH (Secure Shell) app, eg a 'terminal'.
 - There's no web-site or other fancy GUI on the CSF - command-line for now.
- **Linux** users - have a *terminal* application by default
 - Start a Terminal and type the following command:

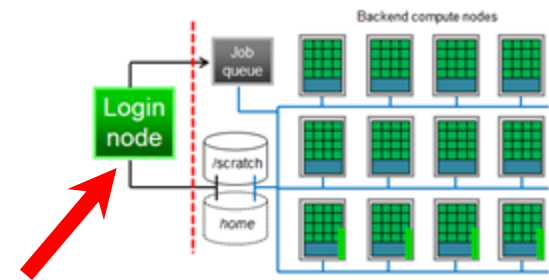
```
ssh -X username@csf3.itservices.manchester.ac.uk
```

UPPERcase X

Central IT Services username.
Answer 'Yes' to continue *if* asked.
Enter central IT password when asked (same as for email)

- Finished using CSF? Log out with: **logout** or **exit**

Connect to CSF from a Mac



- Access the CSF from a PC / laptop using an SSH (Secure Shell) app, eg a 'terminal'.
 - There's no web-site or other fancy GUI on the CSF - command-line for now.
- **Mac users** - have a *terminal* application by default
 - You will need to install X-Quartz first
<https://www.xquartz.org/> (install then you should reboot your Mac)
 - Start a *Terminal* app and type the following command:

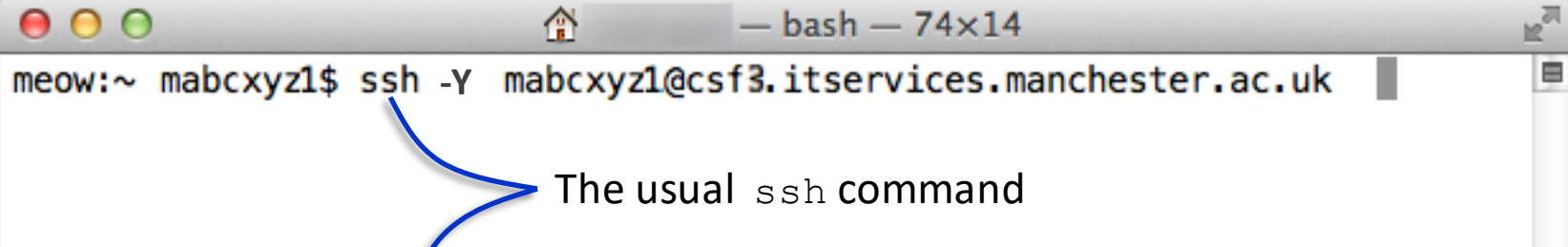
```
ssh -Y username@csf3.itservices.manchester.ac.uk
```

UPPERcase Y

Central IT Services username.
Answer 'Yes' to continue *if* asked.
Enter central IT password when asked (same as for email)

- Finished using CSF? Log out with: **logout** or **exit**

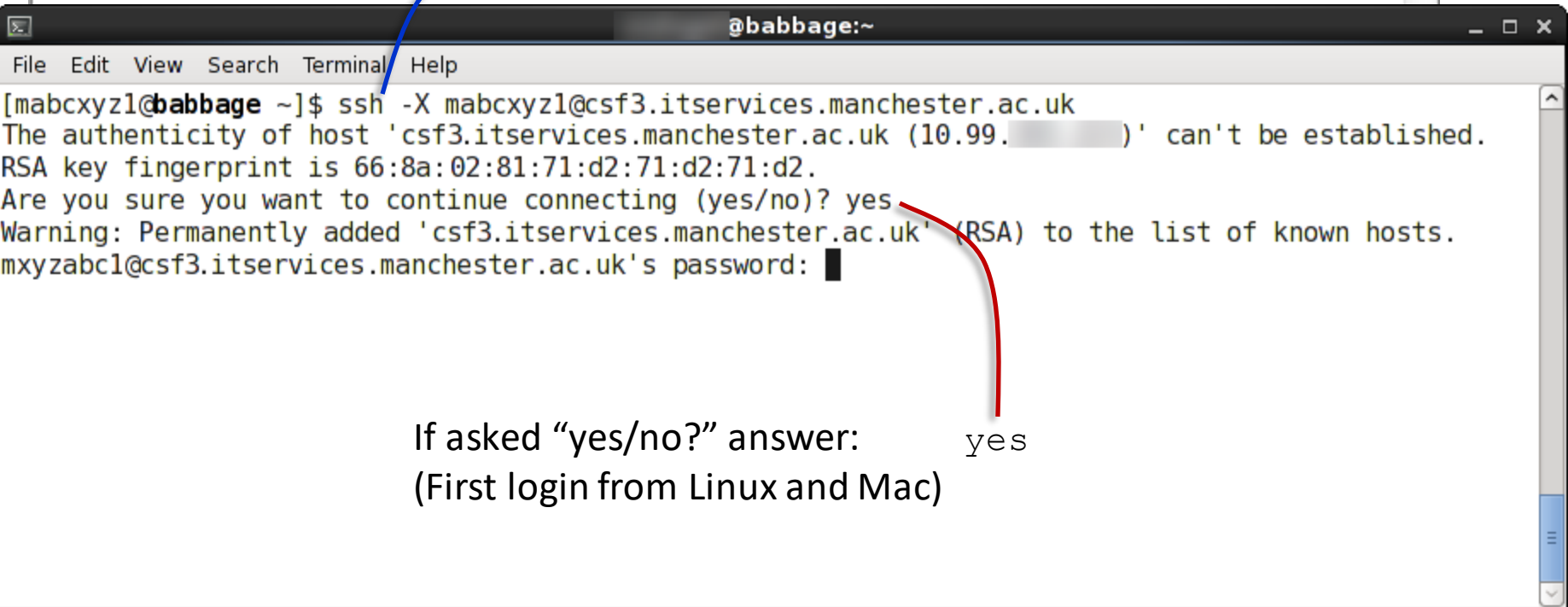
Linux / Mac Terminals



meow:~ mabcxyz1\$ ssh -Y mabcxyz1@csf3.itservices.manchester.ac.uk

The terminal window has a title bar with three colored buttons (red, yellow, green) and a home icon. The title text is "— bash — 74x14". The command is entered at the prompt "meow:~ mabcxyz1\$".

The usual `ssh` command



@babbage:~

File Edit View Search Terminal Help

[mabcxyz1@babbage ~]\$ ssh -X mabcxyz1@csf3.itservices.manchester.ac.uk

The authenticity of host 'csf3.itservices.manchester.ac.uk (10.99.)' can't be established.
RSA key fingerprint is 66:8a:02:81:71:d2:71:d2:71:d2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'csf3.itservices.manchester.ac.uk' (RSA) to the list of known hosts.
mxyzabcl@csf3.itservices.manchester.ac.uk's password: █

The terminal window has a title bar with standard window controls. The title text is "@babbage:~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command is entered at the prompt "[mabcxyz1@babbage ~]\$". The output shows the host's authenticity warning and the user's response "yes".

If asked "yes/no?" answer: yes
(First login from Linux and Mac)

DUO 2FA (when on-campus)

- When on-campus, after you enter your password, all login methods will then ask about DUO:

```
Duo two-factor login for mabcxyz1
```

```
Enter a passcode or select one of the  
following options:
```

```
1. Duo Push to +XX XXXX XX7890
```

```
Passcode or option (1-1): 1
```

```
Success. Logging you in...
```

```
(the message of the day is now displayed)
```

```
[mabcxyz1@login1 [csf3] ~]$
```

Type 1 (and press Enter) in your ssh app to generate a DUO *push* to your device.

Then **accept** the push on your device.


You are now logged in.

What you see when you log in

- CSF uses CentOS Linux (c.f. Red Hat EL)
 - Command line – **requires the input of commands**, can be a little scary at first to new users
 - A welcome *message of the day* - announcements
 - The system awaits input/commands from you at a *prompt* (after you've logged in):

```
[username@login1 [csf3] ~]$  
or [username@login2 [csf3] ~]$
```

- Learning Linux commands (more later):
 - <https://www.chm.bris.ac.uk/unix/>



Type your
commands
at the prompt

Security (2)

- It is **NOT** permitted to share your CSF account
- CSF uses your **IT password** – i.e. same as needed to access UoM email, ...
 - NEVER share it with ANYONE, including IT staff and your supervisor
 - Forgotten it? You can reset it via the IT Account Manager. Will affect all systems that require it.
 - <https://iam.manchester.ac.uk/>
- Reminder: Other general safety measures
 - Install a virus scanner
<https://www.itservices.manchester.ac.uk/cybersecurity/advice/virusprotection/>
 - Be aware of phishing emails
<https://www.itservices.manchester.ac.uk/cybersecurity/advice/phishing/>

PRACTICAL SESSION 1

Logging in

Practical Session 1 – Logging in

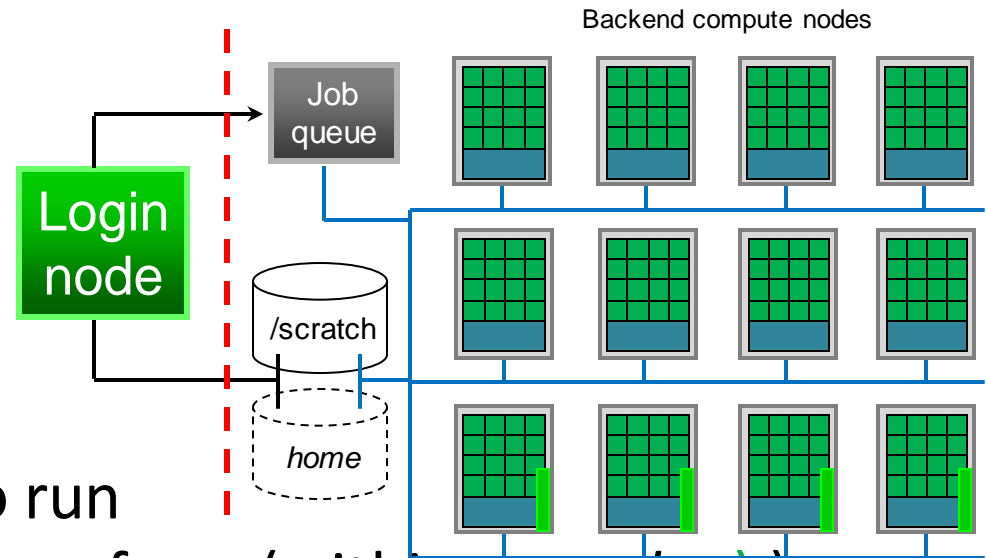
- Exercise 1 sheet (pdf) available at:
<https://ri.itservices.manchester.ac.uk/course/rcsf/>
- Tip: During login, when you are prompted for your password, type carefully – you will not get a cursor that moves or display any *** as you press the keys. But it IS noticing what you type!
- Tip: Once logged in to the CSF, Linux does not always display something after you have entered a command – this is usually a good sign – your command worked, or there was nothing for it to do. If you've got something wrong, it will *usually* tell you via an error message.
- By the end of this practical session, everyone should have successfully logged in to the CSF!
- PLEASE ASK FOR HELP IF YOU RUN IN TO ANY PROBLEMS – WE ARE HERE TO HELP!

RUNNING JOBS

Doing real work on the CSF

Jobs, Jobscripts and the Batch System

- We want to do computational work - “jobs”



- You decide:
 - What program(s) to run
 - Which directory to run from (within *scratch* :-))
 - What resources it needs (#cores, CPU type, memory)
- Write these requirements in a *jobscript*
- Submit your jobscript to the batch system (SGE)
- SGE decides exactly *when* and *where* the job runs

Warning: The login nodes

- Do *not* run computational work here:
 - Not enough cores
 - Not enough memory
 - 100+ users connected, so running work causes serious problems
- You *can* do the following:
 - Transfer files on and off the CSF
 - Set up and submit your jobs (covered in next few slides)
 - Basic data processing/viewing
- Computational work running on the login node will be killed without warning!

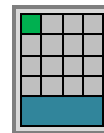
Reminder: Creating Jobscript files (1)

- You need to be able to create a small **text file** to describe your job
- Run `gedit` on the CSF login node - a simple text editor
 - Creates and saves the file *on the CSF*
 - `gedit` is similar to notepad (other Linux editors: emacs, vi)
- Once logged in to the CSF type:
`gedit &` '&' allow you to carry on using the command-line. Try it without to see.
 - Navigate to a file or start typing and then save

Can I Write Jobscripts on Windows?

- A warning about Windows text files (EG: in *notepad*)
 - There's an inconsistency over the (hidden) *end-of-line* characters in text files:
 - Windows: CR (carriage return) + LF (line feed)
 - Linux/Unix: LF (line feed)
 - The extra CR from Windows is a problem in jobscripts. Causes your job to **fail** immediately.
- Solutions
 - Use `gedit` on the CSF login node (writes Linux text files)
 - Or use *notepad*, upload then run `dos2unix myfile.txt`
 - Use only on jobscripts. **Do not come to rely on this – it is too easy to forget to do it – use gedit!**

A simple Jobscript – *Serial* (1 core)



#! on first line only (a special line)

First line indicates we use the *bash* script language to write our jobscript.

#\$ indicates a **batch system parameter** to specify our job requirements. We'll use various combinations of these.

lines are just comments - anything on the line after it will be ignored.

Actual Linux commands we run in our job. They will execute on a compute node.

myjob.txt

```
#!/bin/bash --login
#$ -cwd
#$ -N myjob
#$ -l resource

# Let's do work
date
hostname
sleep 120
date
```

-cwd indicates we'll run from our current (working) directory. Input / output files will usually be found here.

-N (optional). Set the *jobname*. Otherwise will use name of your jobscript as the name.

-l (optional) used to add extra resource requirements e.g. memory, time limits
#\$ -l course only works on the day of a course.

Submit Jobscript to Job Queue

- Submit the jobscript from the login node with:

```
qsub jobscript          # EG: qsub myjob.txt
```

- You will be given a unique *JobID* (6/7-digit number)

```
Your job 598052 ("myjob.txt") has been submitted
```

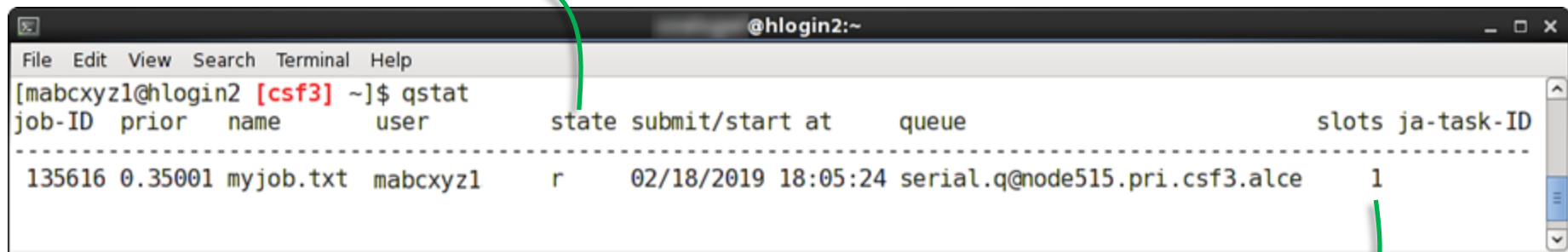
- You can then:
 - carry on with other work, submit other jobs that run other applications without disturbing previous jobs
 - log out of the CSF and your jobs will still run

Check status of your jobs

- To see your job(s) in the batch system, run:

`qstat`

state is either queued (`qw`), running (`r`) or error (`Eqw`)

A terminal window titled '@hlogin2:~' showing the output of the 'qstat' command. The output is a table with columns: job-ID, prior, name, user, state, submit/start at, queue, slots, and ja-task-ID. A single job is listed with ID 135616, priority 0.35001, name 'myjob.txt', user 'mabcxyz1', state 'r', submit time '02/18/2019 18:05:24', queue 'serial.q@node515.pri.csf3.alce', 1 slot, and ja-task-ID 1. A green line points from the 'state' column to the text 'state is either queued (qw), running (r) or error (Eqw)'. Another green line points from the 'slots' column to the text 'slots is the number of cores (1 by default - a serial job)'.

job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
135616	0.35001	myjob.txt	mabcxyz1	r	02/18/2019 18:05:24	serial.q@node515.pri.csf3.alce	1	

slots is the number of cores
(1 by default - a *serial* job)

Serial (1-core) Job Properties

- Our simple example:
 - Serial (unless specified, only 1 core is used - see later)
 - Standard memory (*no* `#$` line asking for more)
 - We get: 4GB RAM (per core - we're just using one core!)
 - Standard 7 day runtime (*no* `#$` line asking for *shorter*)
- Default: **standard serial jobs** will be placed on:
 - Intel nodes (Ivybridge or Haswell)
 - The system looks for a free core on a list of compute node
- We didn't use any: `#$ -l option`
 - EG: `#$ -l mem256` to land on higher memory node

Optional Serial Job Resources

<https://ri.itservices.manchester.ac.uk/csf3/batch/serial-jobs>

Jobscript flag	Description	
-l haswell	5GB/core haswell	NB: This flag is not so useful. The other flags below <i>are</i> .
-l short	1 hour runtime limit , test or pre/post proc	
-l mem256	16GB/core haswell (default)	
-l mem512	32GB/core ivybridge or haswell (system will choose)	
-l mem512 -l ivybridge	32GB/core ivybridge	
-l mem512 -l haswell	32GB/core haswell	

- **7-day runtime limit** on jobs unless otherwise indicated in table.
- Our simple jobscript did *not* use any of the above. Not needed in most cases.
- If you limit a job by *node-type* or memory it may **wait longer in the queue**.
- You will see that the example jobscripts in the exercises have **#\$ -l course**
 - **Only for use today** (reserved nodes on a teaching day).
 - **Remove** if you are practising outside of today.
 - **Never** use on your regular jobs (they'll wait forever!)

So where did my results go?

- If `qstat` returns no output - means job has finished!
- Three possibilities:
 1. If app prints to screen: A text file called ***jobname.oJobID***
 - *Jobname* is the name of your jobscript script (or -N *name* setting)
 - *JobID* is the number of your job
 - Previous example: `myjob.txt.o598052`
 - (Technically: "the std output stream is redirected to the file")
 2. An output file specific to your application
 3. Your job had a problem or failed: it may be reported in one of the above files or in ***jobname.eJobID***
 - (Technically: "the std error stream is redirected to the file")
- Various options to view the file (they are plain text):
 - `cat filename`
 - `less filename` (allows you to page through with spacebar)
 - `gedit filename` (not recommended if it is large)

Why is my job still waiting?

- Your job will **wait** until there are cores available (meeting your jobscript's requirements).
- All jobs can run for up to 7 days (wallclock)
 - (**#\$ -l short** jobs up to 1 hour)
 - Jobs are starting and finishing all the time
- Initially frustrating (perhaps) but **advantages**:
 - You can log off, switch off your PC and your job will stay on the CSF. Log in later to check on job / collect the results.
 - You can submit many jobs.
 - They might not all run at once but SGE will decide...
 - More than one may run at the same time (make sure you have different files/folders for each job).

Many Users Sharing the CSF

- 100s of users running 1000s of jobs
- SGE gives each job a priority: depends on
 - Size of research group's / school's CSF contribution
 - Amount of work already put through by that group and by you as an individual (this month)
- The time for your job to start depends on
 - Priority
 - Availability of requested resource (is CSF busy?)
- Jobs submitted *after* yours may start *before* yours!
- A few Jobs may **never** start
 - SGE tries to spot errors in jobscripts when you run qsub
 - Some may still get through then never run
- We try to ensure that if you submit some jobs, some of them will start within **24 hours**.
 - We make a check every morning of the waiting jobs

How busy is it?

- The system is usually **very** busy.
- However, jobs frequently finish, allowing waiting ones to start
- To see all the jobs for everyone

```
qstat -u "*"
```
- Note: all jobs shown as one list by qstat
 - It displays running and waiting jobs
 - Your job is not necessarily stuck behind all others above yours in the qstat output.
 - CSF is split into a few chunks – the very big jobs do not compete with the smaller jobs for cores
- **Do not try to guess when is a good time to submit your jobs.**
 - If you have work ready to run, **submit** it
 - If your jobs are not in the queue the scheduler cannot consider them
 - **You will waste time, not gain it, by not submitting**

What does Eqw mean? How do I delete a job?

- `qstat` reports your job as 'Eqw'
 - System tried to *start* it, but something went wrong
 - Usually no output file (job hasn't run) to indicate what happened, use:

```
qstat -j JobID | grep error
```
- Error can be cryptic. Most common causes:
 - Missing directory (cannot chdir to ...)
 - You created your jobscript on windows (exec error)
 - Unusual characters or **spaces** in file and directory names
 - No disk space on the filesystem - did you forget to use scratch?
- Detailed advice:
<https://ri.itservices.manchester.ac.uk/csf3/batch/job-monitoring>
- To delete an Eqw job (it'll never run), or one you simply no longer want:

```
qdel jobID
```

PRACTICAL SESSION 2

Serial job

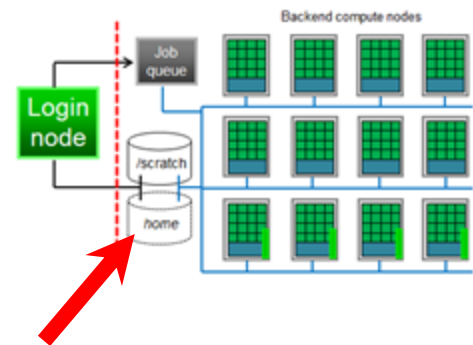
Practical Session 2 - Submitting jobs

- Follow the handout 'Practical Session 2'
 - Use `qsub` to submit a simple serial job on the CSF
 - Use `qstat` to look at the queues
 - Use `qdel` to kill jobs
 - Use `qacct` to look at finished jobs
- Exercise sheet (pdf) available at:
<https://ri.itservices.manchester.ac.uk/course/rcsf/>

CSF STORAGE (FILESYSTEMS)

Where to store your files...

Storage – *Home* filesystem



- Upon login, automatically placed in your *home* directory (folder)
- Limited space, quota shared by everyone in the group
- Uses the Research Data Service (networked storage)
 - Large files can be slow-ish to read/write (implications for jobs)
- Which directory (folder) am I currently in?

```
pwd
```

- How much space am I using? (Linux commands!)

```
du -sh dirname # Can take a while
```

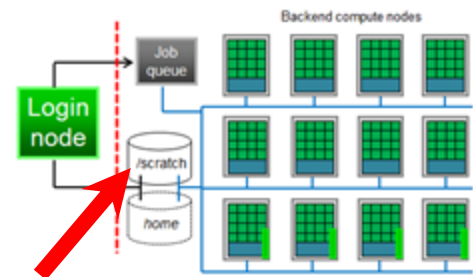
- How big is that file?

```
ls -lh filename # Letter e! not number 1
```

- How much space is used/free overall?

```
df -h . # The . is important!
```

Filesystems - Scratch (1)



- Filesystem local to CSF for:
 - Temporary files - can be huge
 - Running jobs from (it is faster!). Recommended!
- Shared by all CSF users, but we have 1.2PB
- Tidy up after each job finishes
- **Clean-up policy applies: files that have not been accessed for the past 3 months may be deleted automatically**
- **Not backed up!**
 - Move/copy important results to *home* area
 - Not considered safe for long term storage - hardware failure could cause data loss

Storage - *Home* filesystem

- *Home* is **backed up** and **mirrored** to another datacentre
 - Keep **important** files here (results, jobscripts, source code, ...)
 - Deleted a file by mistake? its-ri-team@manchester.ac.uk can tell *you* how to retrieve it
- Only you can access your home directory
 - File permissions can be used to give others access
 - Contact its-ri-team@manchester.ac.uk if you want advice on this as they can be complex
- **Do not** run jobs from your *home* area (see later)
 - Can generate a lot of files, some of them large
 - Using up all of the shared space will make your colleagues unhappy!
 - Consider compressing large (text) files with `gzip`

Filesystems - Scratch (2)

- Using scratch is easy: after log in, change to it:

```
cd ~/scratch
```

- Uses a 'symlink' (short cut) in your home dir to
/scratch/username

- Create a directory (now we're in scratch):

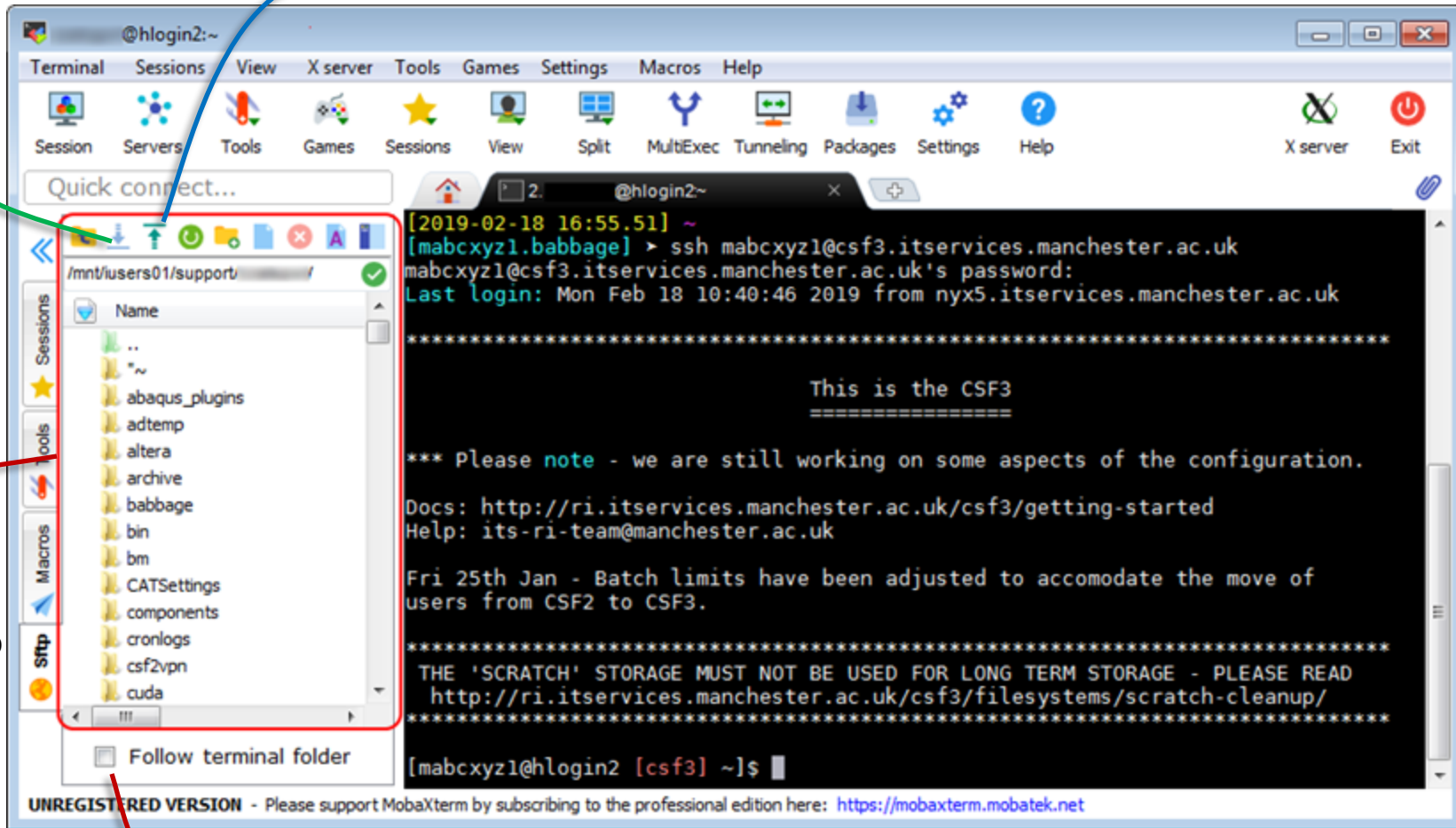
```
mkdir myjobdir
```

- Put all files relevant to your job in that directory and *run your jobs* from there - we'll try this out soon...
- **All compute nodes** see the same scratch area

File transfer with MobaXterm

2. First select files in the MobaXterm browser. Then the **Download** button opens a file-browser to select a destination folder on your PC.

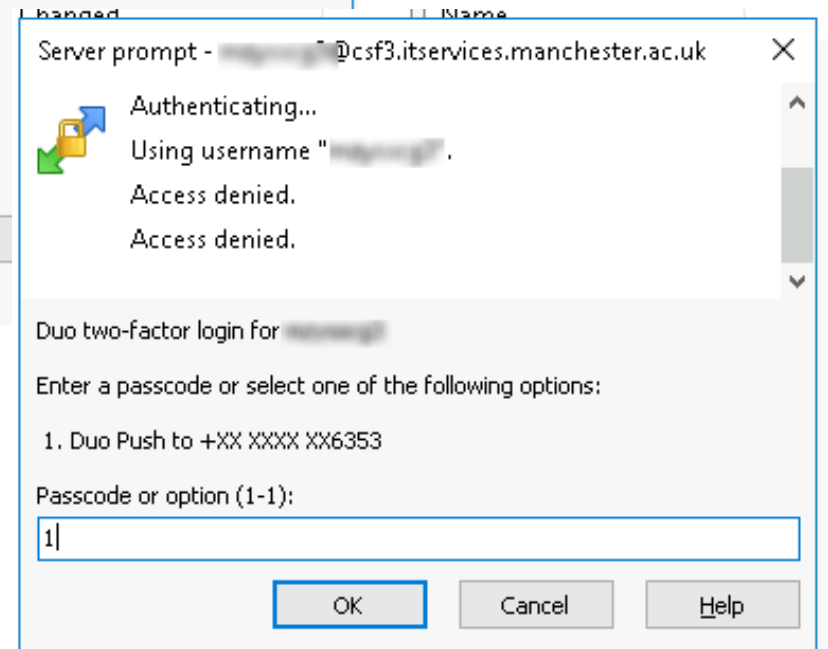
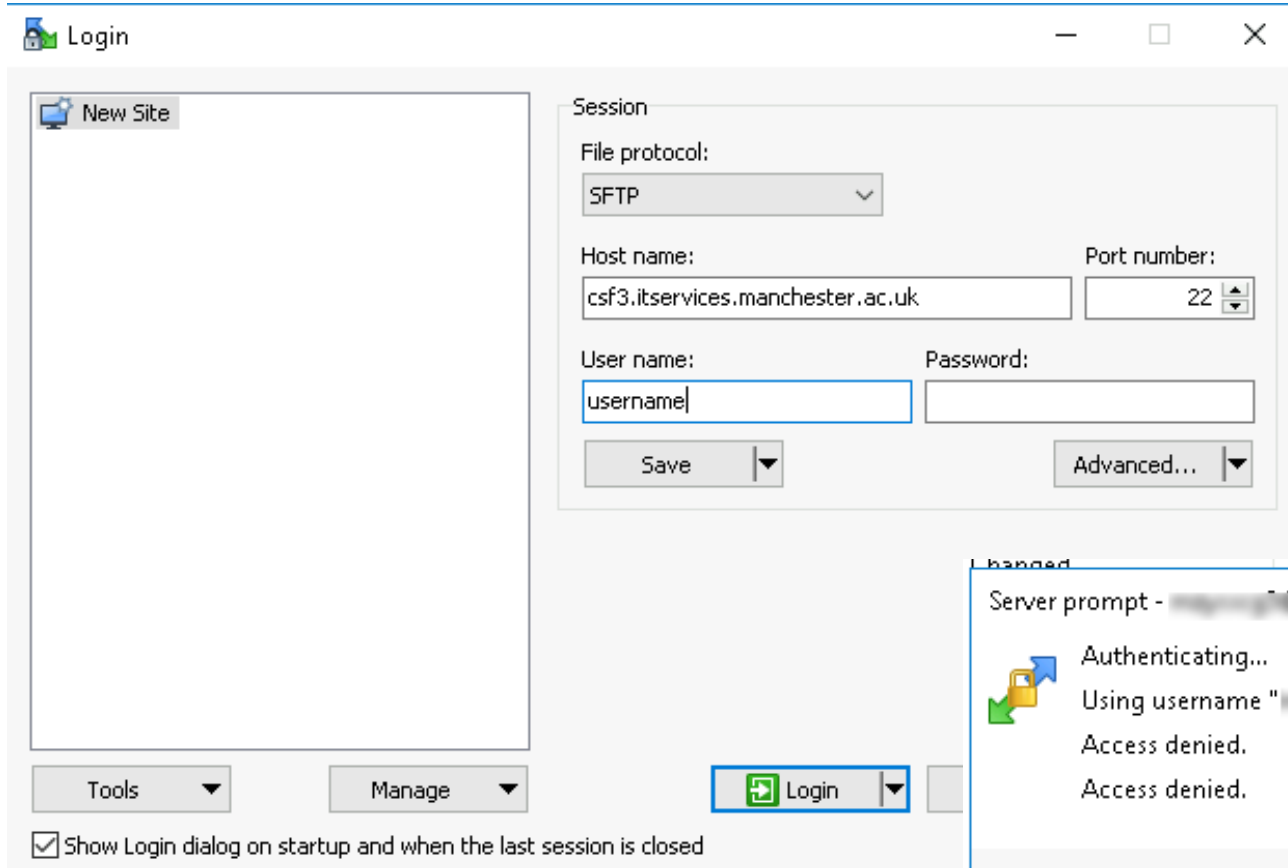
3. The **Upload** button opens a file-browser to select files on your computer to **upload to** CSF (current directory).



1. Drag-n-drop
files from
Windows
Explorer or
Desktop
to/from here.

This can be flaky - do not use

File transfer with WinSCP



File transfer with WinSCP

@csf3.itservices.manchester.ac.uk - WinSCP

Local Mark Files Commands Session Options Remote Help

Synchronize Queue Transfer Settings Default

@csf3.itservices.manchester.ac.uk X New Session

My documents Upload Edit Properties New Download Edit Properties New Find Files

C:\Users\...\Documents\ /mnt/iusers01/support/

Name	Size	Type	Changed	Name	Size	Changed	Rights	Owner
..		Parent directory	31/10/2022 16:44:51	archive		31/10/2022 09:40:21	rw-r-xr-x	root
MobaXterm		File folder	31/10/2022 16:44:51	bin		06/05/2022 15:21:21	rw-r-xr-x	mzyssc...
WindowsPowerShell		File folder	08/04/2022 11:46:18	Chris		14/06/2022 10:03:46	rw-r-xr-x	mzyssc...
activedirectory_user_s...	3 KB	Windows PowerS...	10/03/2021 15:44:59	Circuitscape-4.0.5		17/03/2021 16:38:16	rw-r-xr-x	mzyssc...
				client-info		07/05/2014 08:47:34	rw-r-xr-x	mzyssc...
				CSF3		24/08/2022 09:22:19	rw-r-xr-x	root
				CSF4		26/10/2022 09:52:41	rw-r-xr-x	mzyssc...
				csf-course		15/06/2022 16:20:16	rw-r-xr-x	mzyssc...
				damBreak		28/01/2022 12:46:14	rw-r-xr-x	mzyssc...
				dbxcli_testdir		06/09/2022 12:01:26	rw-r-xr-x	mzyssc...
				Desktop		28/09/2020 22:46:28	rw-r-xr-x	mzyssc...
				Documents		03/02/2021 17:30:51	rw-r-xr-x	mzyssc...
				Downloads		27/05/2020 16:04:55	rw-r-xr-x	mzyssc...
				errorsClaudio		14/03/2022 12:16:26	rw-r-xr-x	mzyssc...
				example-job		07/09/2022 08:49:59	rw-r-xr-x	mzyssc...
				external		24/08/2022 09:32:38	rw-r-xr-x	root
				foam		07/04/2022 11:06:54	rw-r-xr-x	mzyssc...
				fromChris-forWendy		04/10/2022 12:46:14	rw-r-xr-x	mzyssc...
				ge-duetto		16/02/2022 21:47:30	rw-r-xr-x	mzyssc...
				go		28/05/2020 15:50:07	rw-r-xr-x	mzyssc...
				gridware		12/05/2022 14:24:53	rw-r-xr-x	mzyssc...
				HPC_Pool		16/09/2019 10:18:37	rw-r-xr-x	mzyssc...
						25/11/2021 10:24:07	rw-r-xr-x	mzyssc...

0 B of 2.01 KB in 0 of 3 5 hidden 0 B of 10.2 GB in 0 of 116 127 hidden

SFTP-3 0:05:44

home file transfer with Linux / Mac

(can also type in a local MobaXterm window)

The : is
important!

- Transfer a file from your computer to your CSF home dir

```
scp myfile.txt username@csf3.itservices.manchester.ac.uk:~/training/
```

Exercise: Create a file on your PC named myfile.txt containing some text then transfer it to the CSF.

Destination directory on the CSF
~ is shorthand meaning "your home directory"
If no destination after the : then uses "your home directory"

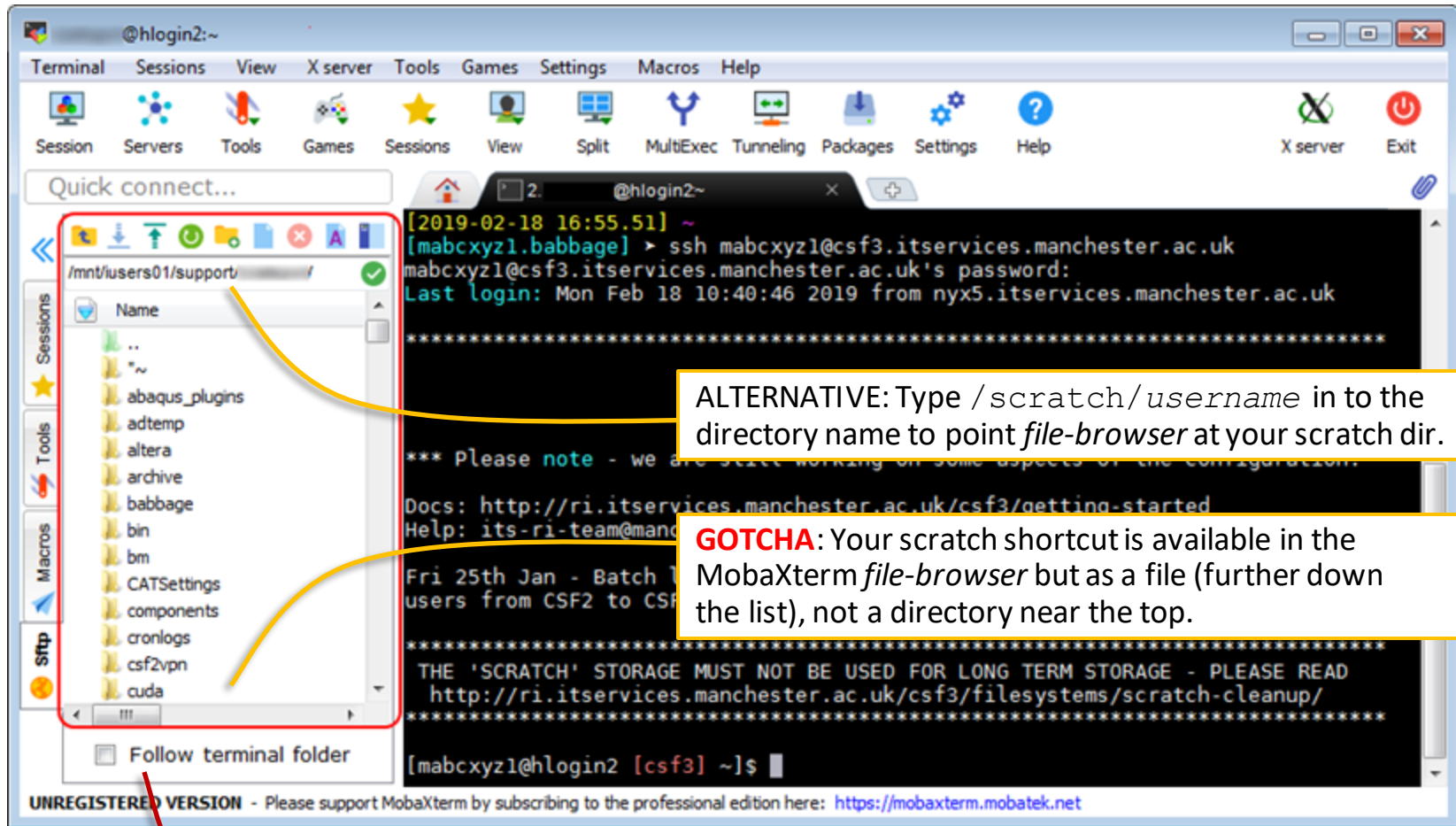
- Transfer a file from your CSF home dir to your computer

```
scp username@csf3.itservices.manchester.ac.uk:results.out .
```

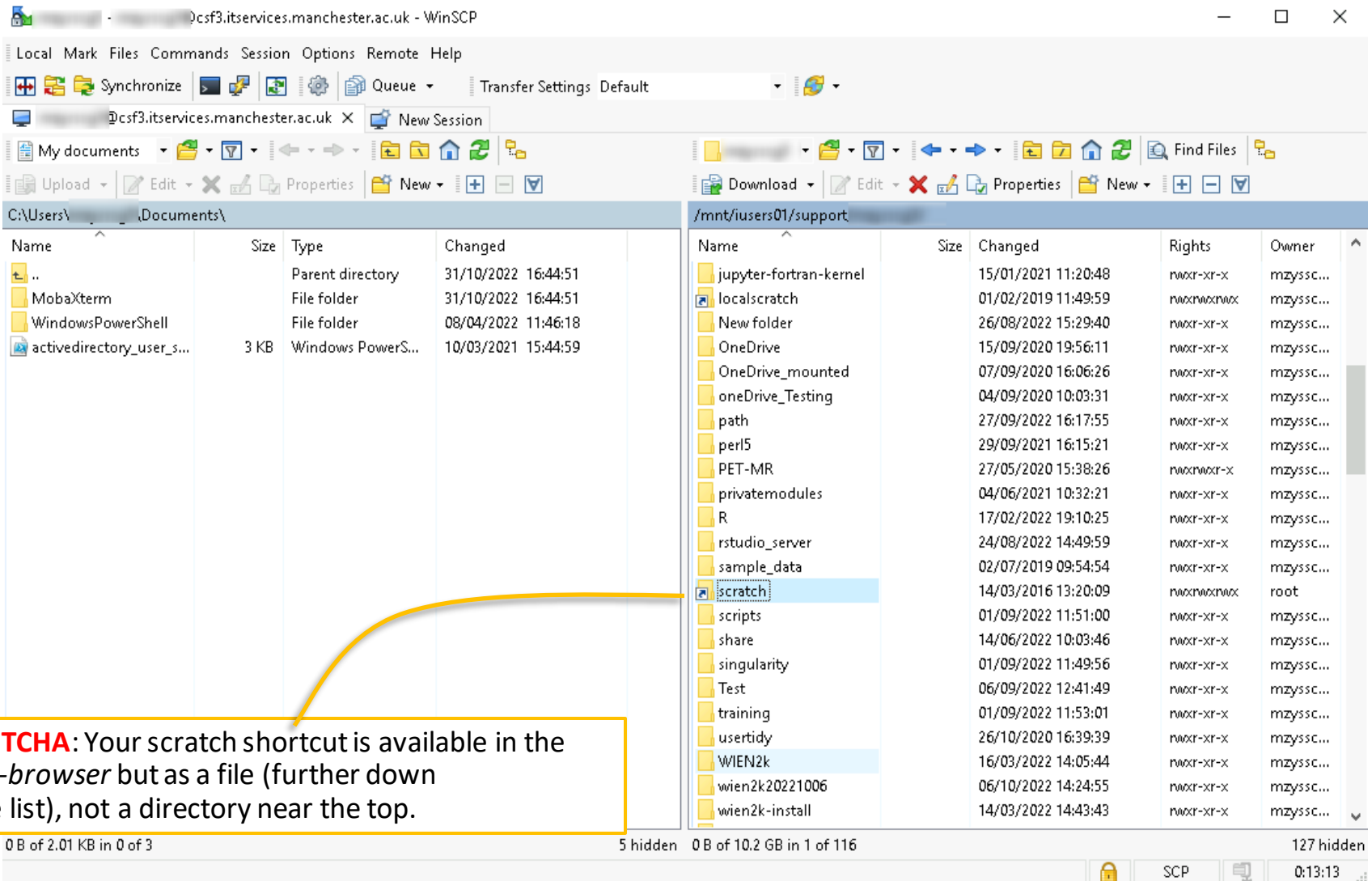
The . is shorthand meaning
"the current directory" on **your** computer

- Change directory & filenames...

Accessing *scratch* with MobaXterm



Accessing *scratch* with WinSCP



WinSCP interface showing local and remote file systems. The local pane shows 'C:\Users\...Documents\' and the remote pane shows '/mnt/iusers01/support'. The 'scratch' file is highlighted in the remote list. A callout box points to 'scratch' with the text: 'GOTCHA: Your scratch shortcut is available in the file-browser but as a file (further down the list), not a directory near the top.'

Name	Size	Type	Changed
..		Parent directory	31/10/2022 16:44:51
MobaXterm		File folder	31/10/2022 16:44:51
WindowsPowerShell		File folder	08/04/2022 11:46:18
activedirectory_user_s...	3 KB	Windows PowerS...	10/03/2021 15:44:59

Name	Size	Changed	Rights	Owner
jupyter-fortran-kernel		15/01/2021 11:20:48	rw-r--r--	mzyssc...
localscratch		01/02/2019 11:49:59	rw-rw-rw-	mzyssc...
New folder		26/08/2022 15:29:40	rw-r--r--	mzyssc...
OneDrive		15/09/2020 19:56:11	rw-r--r--	mzyssc...
OneDrive_mounted		07/09/2020 16:06:26	rw-r--r--	mzyssc...
oneDrive_Testing		04/09/2020 10:03:31	rw-r--r--	mzyssc...
path		27/09/2022 16:17:55	rw-r--r--	mzyssc...
perl5		29/09/2021 16:15:21	rw-r--r--	mzyssc...
PET-MR		27/05/2020 15:38:26	rw-rw-r--	mzyssc...
privatemodules		04/06/2021 10:32:21	rw-r--r--	mzyssc...
R		17/02/2022 19:10:25	rw-r--r--	mzyssc...
rstudio_server		24/08/2022 14:49:59	rw-r--r--	mzyssc...
sample_data		02/07/2019 09:54:54	rw-r--r--	mzyssc...
scratch		14/03/2016 13:20:09	rw-rw-rw-	root
scripts		01/09/2022 11:51:00	rw-r--r--	mzyssc...
share		14/06/2022 10:03:46	rw-r--r--	mzyssc...
singularity		01/09/2022 11:49:56	rw-r--r--	mzyssc...
Test		06/09/2022 12:41:49	rw-r--r--	mzyssc...
training		01/09/2022 11:53:01	rw-r--r--	mzyssc...
usertidy		26/10/2020 16:39:39	rw-r--r--	mzyssc...
WIEN2k		16/03/2022 14:05:44	rw-r--r--	mzyssc...
wien2k20221006		06/10/2022 14:24:55	rw-r--r--	mzyssc...
wien2k-install		14/03/2022 14:43:43	rw-r--r--	mzyssc...

0 B of 2.01 KB in 0 of 3 5 hidden 0 B of 10.2 GB in 1 of 116 127 hidden 0:13:13

scratch file transfer with Linux / Mac

(can also type in a local MobaXterm window)

- Very similar to commands used earlier for our *home* directory
- Transfer a file from your computer to your CSF scratch dir

```
scp file2.txt username@csf3.itservices.manchester.ac.uk:~/scratch/
```

We give a destination after
the : meaning “use my scratch shortcut”
Omit the destination to transfer to *home* dir

- Transfer a file from your CSF scratch to your computer

```
scp username@csf3.itservices.manchester.ac.uk:scratch/results2.out results2.copy
```

Now you get a copy with a
different name on your computer.
Use . to keep the same name (results2.out)

- Change directory & filenames as required...

Extra Storage Space (Optional)

- Some research groups have extra space, example path:

`/mnt/eps01-rds/group/username`

- No shortcut from your home? To access it use:

`cd /mnt/eps01-rds/group/username`

- To create a shortcut (named `data`) in your home area:

`cd ~`

`ln -s /mnt/eps01-rds/group/username data`

- Also backed up
- Often many TB, but again shared by everyone else from your group
 - Be fair in your usage

Additional filesystem/file transfer info

- We have additional info about how to manage your files and your disk usage:
<https://ri.itservices.manchester.ac.uk/userdocs/file-management/>
- Docs about file transfer:
<https://ri.itservices.manchester.ac.uk/userdocs/file-transfer/>
- If you need to transfer a lot of files or big files to and from the CSF please do not do it on the login node
 - Ask for an account on the **RDS-SSH service**
- Many file management tasks can be included in your batch jobs – see the FAQ.

Basic Linux File Commands

A good Linux tutorial is available at: <https://www.chm.bris.ac.uk/unix/>

Command	Description
<code>less file1</code> <code>zless file2.gz</code>	Display the content of <code>file1</code> (text file) a page at a time on screen. If you've compressed <code>file2</code> with <code>gzip</code> , no need to uncompress first. Press <code>space</code> to page down through a long file Press <code>return</code> to scroll down a line at a time Press <code>b</code> to scroll back up a page Press <code>G</code> to go to end of file Press <code>q</code> to quit/exit
<code>cat file1</code> <code>zcat file2.gz</code>	Dump entire file to screen (a quick way to look at text files). If you've compress <code>file2</code> with <code>gzip</code> , no need to uncompress first.
<code>gedit file1</code>	Edit <code>file1</code> using a simple graphical text editor (similar to notepad on Windows). See later for more on opening graphical programs on the CSF so that they display a window on your computer.
<code>file filenameA</code>	Try to tell us what type of data is in <code>filenameA</code> . Useful to determine the output of some program where you are not sure what type of output it has generated. For example: <code>file output.dat</code> Might be <code>ASCII text</code> (so we can look at it with <code>less</code> or <code>gedit</code>) or might be <code>data</code> (you'll need some other program to read it)
<code>du -sh .</code>	How much disk space is current directory (all files and subdirs) using?
<code>df -h .</code>	How much free space is there in the current area?

Basic Linux File Commands

A good Linux tutorial is available at: <https://www.chm.bris.ac.uk/unix/>

Command	Description
<code>cd dir1</code> <code>cd ~/dir1/dir2</code> <code>cd ..</code> <code>cd</code>	Change directory (go in to <code>dir1</code> which is located in the current dir) Go in to <code>dir2</code> in <code>dir1</code> in home (~ is shorthand for <i>home</i>) Go up to parent directory (e.g., from <code>~/dir1/dir2</code> to <code>~/dir1</code>) Go back to <i>home</i> (useful if you become lost)
<code>pwd</code>	Lost? Print Working Directory (display current location)
<code>ls</code> <code>ls -lh</code> <code>ls -lh file1 dirA</code> <code>ls -lh dirA/*.dat</code>	List content (names of files and directories) of current directory List in long form (dates, file sizes, names) current directory List in long form (dates, file sizes, names) specified files, directories ... List in long form all files ending in <code>.dat</code> in directory <code>dirA</code>
<code>mkdir dirA</code>	Make directory named <code>dirA</code> (in the current directory)
<code>cp fileA fileB</code>	Copy (duplicate) a file (copy <code>fileA</code> to a new file <code>fileB</code>)
<code>mv fileC fileD</code> <code>mv fileE dirA</code> <code>mv fileF dirA/fileG</code>	Rename a file (from <code>fileC</code> to <code>fileD</code>). Works for directories too. Move <code>fileE</code> in to sub-directory <code>dirA</code> (<code>dirA</code> must exist) Move <code>fileF</code> AND rename it all in one go (<code>dirA</code> must exist)
<code>rm fileH</code>	Delete (remove) a file (caution!!)
<code>rm -rf dir1</code>	Delete directory and all files (and other sub-dirs) in there (caution!!!!!!)
<code>gzip bigfile</code> <code>gunzip bigfile.gz</code>	Compress a file (becomes <code>bigfile.gz</code>) to make better use of disk-space. Text files usually compress well. Uncompress previously compressed file (becomes <code>bigfile</code>). 69

PRACTICAL SESSION 3

File Transfer

Practical Session 3 – File Transfer

- Follow the hand-out 'Practical Session 3'
 - Transfer a file from the CSF to your PC
 - Transfer a file from your PC to the CSF
 - Windows: use MobaXterm, Mac/Linux: use "scp"
 - Or, if time permits, Windows users can try option 2 - <https://winscp.net/eng/download.php>
- This is not a 'real' world example, but:
 - You may need to generate files on your PC for processing on the CSF (e.g. an "abaqus" input file)
 - Your supervisor may give you files that you then need to transfer to CSF
- Exercise sheet (pdf) available at: <https://ri.itservices.manchester.ac.uk/course/rcsf/>

Need more help with the CSF?

- Extensive documentation about all aspects of the service:
<https://ri.itservices.manchester.ac.uk/csf3/>
- Contact the Research Infrastructure Team
its-ri-team@manchester.ac.uk
- See you:
 - After lunch (in-person courses)

Thank you!