Data management on HPC platforms

Transferring data and handling code with Git

1 What tool for what data

Exercise 1: Simple connection

To connect to the front node of a cluster if it is your first time.
On Linux/Mac open a terminal and use `ssh <username>@<machine>`
On Windows use a soft like PuTTY or Tunnelier

- Tunnelier https://bitvise.com/tunnelier
- PuTTY http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Front nodes:
- bellatrix
- castor
- deneb1, deneb2

- Connect to your favorite front node
- Check the different folders `/home` `/scratch` `/work`
- Ok this exercise is just to be sure you can connect to the cluster

2 Big data, single user: old style rocks

Exercise 2: Using `scp`

How to use `scp` / `pscp.exe`
Send data to remote machine:
`scp [-r] <local_path> <username>@<remote>:<remote_path>`
Retrieve data from remote machine:
`scp [-r] <username>@<remote>:<remote_path> <local_path>`

- Copy a file from your machine to the cluster.
- Retrieve a file from the cluster to your machine

Exercise 3: Using `sftp`

Filezilla https://filezilla-project.org
Cyberduck https://cyberduck.io/
Tunnelier https://bitvise.com/tunnelier

- Try downloading one of these tools and connect to a cluster

Exercise 4: Using `rsync`

Mode infos on rsync:
- `rsync -auvP <src folder> <dst folder>`
- `rsync -auvP <remote>:<src folder> <dst folder>`
- `rsync -auvP <src folder> <remote>:<dst folder>`
- `man rsync`

- Create a temporary `tmp/` folder in your home folder
- Copy the folder `/ssoft/sources/cmake-3.2.3` to this `tmp/` folder
- Create a `backup/` folder in the `tmp/` folder
- Use rsync to copy the `cmake` folder in your `tmp/backup` folder
- Modify the `README` file in the code of cmake in `tmp/cmake-3.2.3`
- Re-synchronize the files

3 Small data any number of users: versioning is the key

Exercise 5: First step with Git

- If you do not have git installed, get it from https://git-scm.com/downloads or from your package manager
- Go on https://c4science.ch/ and login with your EPFL account (Login for Swiss Universities).
- Once connected go on the setting page (the wrench on the top right corner)
- In the Authentication > VCS Password menu set a password. This password will be used to connect to the git server through https.

Exercise 6: First step with Git

`git clone <repo url> [local name]`	Clone a remote repository
`git add <files...>`	Stage modified/new files
`git commit -m "comment"`	Commit staged files
`git pull`	Pull and merge remote modifications
`git push`	Push the local modifications to the remote server
`git status`	Check the local state

- Now you should be able to clone the repository
`https://c4science.ch/diffusion/SCTESTREPO/scitas-test-repo.git`
- Create a file, use a filename that will not clash with the others
- Check the state of your working copy
- Add the file to the repository
- Commit your modifications
- Pull the potential modifications from the server
- Push your changes to the server

Exercise 7: Generate and solve conflicts

- Clone the `scitas-test` repository in a different folder
- Modify the file created in the previous exercise in both clones
- Commit this both modifications
- Pull and push in one of the clone
- Pull in the second clone, You should get a conflict

```
<<<<<<<<<<
One version
==========
Other version
>>>>>>>>>>
```

- Check the local status
- Correct the conflict and commit using `git commit -a`
- Push the modifications

Exercise 8: Branches/merges

`git branch <name>`	Create a new branch from the current HEAD
`git checkout <name>`	Switch to the specified branch
`git merge <name>`	Merge the branch specified in the current one
`git branch -d`	Delete a branch
`git branch -a`	List all branches
`git log`	List the different commits of the current branch
`git log --graph --all`	Show also the branches

- Create a branch with the name of your choice
- Modify a file and commit the changes
- Checkout the `master` branch
- Modify a file and commit the changes
- Merge the branch previously created in the `master` branch
- List all branches
- Print the logs of the different modifications
- Delete the merged branch

Exercise 9: Handle remotes

`git init --bare`	Create a new server
`git remote add server <url>`	Add a remote server
`git remote -v`	Show the remotes configured
`git push <remote name>`	Push to a given remote

- Connect on the front node of your favorite cluster
- Create a new folder that will contain your server
- In this folder initialize a new git server
- In one of the former clone of `scitas-test` add the new remote URL `<cluster name>:<path to repo>`
- List the remotes to see if everything looks correct
- Push the local content to the new server
- On the cluster clone this new server URL `<path to repo>`
- Note: The access permission on this new server are based on the file system permissions