

# Accessing PadLabNas

Daniel Padfield

July 7, 2025

## Table of contents

<b>An introduction to using PadLabNas</b>	<b>2</b>
General Rules . . . . .	2
Folder and file names . . . . .	2
Folder structure . . . . .	3
Permitted uses of PadLabNas . . . . .	3
Back-up OneDrive (or other storage service) using CloudSync . . . . .	3
Back-up data . . . . .	4
Moving files to and from PadLabNas . . . . .	4
File Station . . . . .	4
Download Station . . . . .	4
Cloud Sync . . . . .	4
rclone . . . . .	5
Share folders with a Share Link . . . . .	7

# An introduction to using PadLabNas

PadLabNas is a Network Attached Storage designed to simplify how members of the Buckling Collaborative backup, archive, and share their data. It lives in Dan Padfield's office.

Log in to PadLabNas from <https://padpadpadpad.de6.quickconnect.to/>. Request access by contacting Daniel Padfield at [d.padfield@exeter.ac.uk](mailto:d.padfield@exeter.ac.uk) or messaging him on Microsoft Teams.

## General Rules

- Set schedule settings for syncs to exclude 8-5 Monday to Friday so it does not read/write during work hours when people are in the office.
- Every folder within **BucklingCollaborative** has to have a README describing the folder.
- If moving files between **PadLabNas** and another remote server, ssh into **PadLabNas**. If you try using your local machine as a go-between for two remotes using **rclone**, it will download the files to your machine and re-upload them.
- When using **rclone**, make sure to either use **screen** or **tmux** to allow the upload and download to continue even if the Terminal window closes.

## Folder and file names

- How you name folders within your home folder is up to you, as long as it is consistent.
- In **BucklingCollaborative** name folders with underscores. I recommend naming folders by the year of the project and

underscore with a summary title of the project and the initials of the main researcher on the project. For example: “2025\_doseresponseAMR\_ZE”

## Folder structure

- You can view all files you have access to on the Synology by using the **File Station** application.
- Each member has their own home space, which can be seen in **home**, you can see everyone else’s in **homes**.
- Put your own personal files in your space, but any finished projects and sequencing goes into **BucklingCollaborative**.
- Within **Buckling Collaborative**, there are some legacy folders which store previous projects that were backed up in a variety of places.
  - **buckling** stores
  - **padfield-lab** stores
  - **1840-cmh-bacteria-radiations** stores

## Permitted uses of PadLabNas

### Back-up OneDrive (or other storage service) using CloudSync

- **CloudSync** is an application on the virtual Desktop of PadLabNas. If not, you can access it on Package Centre.
- Once there set up your OneDrive account for the Synology,
  - I use Bidirectional Sync Direction, meaning files go from the NAS to your OneDrive, and *vice versa*.
  - Set the local path to be **one\_drive** within your profile area (e.g. dpadfield/one\_drive). Create the **one\_drive** folder if needed.

## Back-up data

- Back up each project into its separate folder in BucklingCollaborative.
- Ensure it has a README.
  - For sequencing, a [template](#) exists on the BucklingMassive Sharepoint.
  - For other projects you can see good advice [here](#) and [here](#). Essentially describe the dataset and each file in the folder and what it does. An example of one of my projects is [here](#).
- Make sure the folder is named consistently (see Section ).
- Depending on where the sequencing is, you may download it in several ways (see below).

## Moving files to and from PadLabNas

### File Station

**Drag and drop** using access from <https://padpadpadpad.de6.quickconnect.to/>. With **File Station** you can add, move, and rename folders, as well as deleting files.

### Download Station

Use the application **Download Station** to download files from third party providers when they give a download link.

### Cloud Sync

Use the application **Cloud Sync** to sync your OneDrive or other cloud storage system regularly to the NAS.

## rclone

Programmatically move files using **rclone**. **rclone** is useful to move individual files and folders between servers or to and from your local machine. If moving to or from your local machine, this can be done on the command line, if moving to or from PadLabNas to another server (e.g. Athena, Carson, Isca, RStudio), you will need to ssh into the other server, and install **rclone** using **conda** in your base environment.

First you will need to set up **rclone**. Open your terminal. On your local machine (and any remote server you are also using), you need to create a *config* file which stores the details of the PadLabNas (and any other remotes) you want to move files between. When moving files between PadLabNas and another remote, you will need to SSH into the remote first and add a config file to the remote server.

```
# ssh in PadLabNas (if needed)
ssh USERNAME@ipaddresshere -p 22

# Check _rclone_ is installed
rclone -version
# if not install rclone (on your local machine)

# add a remote
rclone config
```

This will start an interactive command in your terminal to add a new remote. To set this up:

1. When asked for *Storage>* choose **ssh/sftp (48)**.
2. When asked for *host>* add the address of the remote (e.g. **the ip address**)
3. When asked for *user>* use the user for that remote (e.g. **dpadfield** or **dp415**)
4. When asked for *port>* use **22**

5. When asked for *password*> type your password for that server.
6. All the other options can be left empty or as the default.

To be able to move files to and from PadLabNas easily, we need to manually edit the config file. First we need to know where the config file is.

```
# local rclone config file
rclone config file
```

Will tell you where your config file is stored. Open it. In the [PadLabNas] section add **path\_override = @/volume1/** on a new line. Save and close the file.

We are now ready to move folders and files programmitically. Some useful usecases are below. [rclone](#) docs are useful for writing commands. Chat-GPT or Claude AI may also be useful for this. Some example commands are below. Adding *-dry-run* to your copy commands will give you output of what *rclone copy* will do without moving any files.

```
# list available remotes
rclone listremotes

# view directories of a remote_
rclone ls PadLabNas:

# move to where you want to move the folder too
cd ~/Downloads

# run a test download using copy
# downloads test from PadLabNas into your local machine to a
↳ folder. Adding /test at the end of the destination
↳ creates the folder on your machine.
rclone copy PadLabNas:BucklingCollaborative/test
↳ ~/Downloads/test --progress
```

```
# move test onto PadLabNas
rclone copy ~/Downloads/test --progress
↪ PadLabNas:BucklingCollaborative/test
```

An example of moving a folder from PadLabNas to another remote server is below. This example assumes you are logged in to the remote server (e.g. RStudio server). It is important to write absolute paths and not use ~ to denote relative file paths.

```
# move test folder from PadLabNas to remote server
rclone copy PadLabNas:BucklingCollaborative/test
↪ /home2/ISAD/dp415/test --verbose --progress
```

## Share folders with a Share Link

Sometimes you may want to share a folder with people who are not signed up to PadLabNas. To share a folder, Right Click on the folder in **File Station** and click Share. Then click Save. You can then share that link with anyone you want and they can Download it through their browser. **NOTE the recipient still needs to be on the University VPN for this to work.**

If you want them to be able to download it through a server (e.g. RStudio server) or programmatically generally, the process is slightly more involved. You can download an example script for this [here.](#), but the code is also below. **If it is a folder, make sure it is compressed to a .zip file by Right Click Compress to .zip.** Create a Share Link as above. Send Share link with recipient.

The following code is for the recipient. Open the Terminal Client.

```

# check curl is installed
curl --version

# Move into correct directory to download the file or zip
↳ folder.
cd ~/Downloads

# Define Sharing Link, NAS address, and file name. You may
↳ need to change these bits.

# sharing link_
sharing_link=https://gofile.me/7EFXM/HwTkOww2x

# PadLabNAS address
nas=https://padlabnas.synology.me:5001

# filename
filename="test.zip"

#Create a file that allows curl to talk to the Sharing Link
↳ properly. We call it cookies.txt. You can just copy and
↳ paste this code.
curl -k -c cookies.txt -b cookies.txt
↳ "$nas/sharing/${basename "$sharing_link"}"

#Download the file using curl and remove cookies.txt. Just
↳ copy and paste this code.
curl -k -b cookies.txt -L -O "$nas/fsdownload/${basename
↳ "$sharing_link")/$filename"

# remove cookies file_
rm cookies.txt

# Unzip the folder if you want to!

```



```
unzip "$filename" -d ~/Downloads
```