User Guide

15. August 2022
# Table of Contents

Introduction ......................................................................................................................... 4
Requirements and Terms of Use ......................................................................................... 4
How to Access UCPH ERDA ............................................................................................... 4
  Sign-up ............................................................................................................................... 5
  Login ................................................................................................................................ 8
Overview ............................................................................................................................... 8
Home .................................................................................................................................... 9
Files ..................................................................................................................................... 10
File Sharing and Data Exchange ......................................................................................... 16
  Share Links ......................................................................................................................... 16
  Workgroup Shared Folders ............................................................................................... 20
Workgroups (VGrids) .......................................................................................................... 21
  Workgroup Web Pages ..................................................................................................... 22
Archives ................................................................................................................................ 23
Jupyter ................................................................................................................................... 28
Jupyter Notebook Customization .......................................................................................... 40
  Python ................................................................................................................................ 40
  R / RStudio ....................................................................................................................... 41
  Other Packages and Install .............................................................................................. 41
MATLAB programs on DAG and MODI .............................................................................. 41
  Helloworld example ....................................................................................................... 43
Settings + Setup .................................................................................................................. 44
  2-Factor Authentication .................................................................................................... 46
Logout ................................................................................................................................... 51
Advanced Data Access ........................................................................................................ 52
WebDAVS ............................................................................................................................. 52
  Windows File Manager .................................................................................................... 53
  Adjust Windows WebDAVS Transfer Limit ..................................................................... 58
Mac OSX File Manager ........................................................................................................ 64
  Linux File Manager ......................................................................................................... 66
SFTP ...................................................................................................................................... 68
  FileZilla ............................................................................................................................. 69
  WinSCP .............................................................................................................................. 72
PuTTY SFTP .......................................................................................................................... 72
  Linux File Manager ......................................................................................................... 73
SSHFS .................................................................................................................................... 73
FTPS ..................................................................................................................................... 74
  FileZilla ............................................................................................................................. 75
  WinSCP .............................................................................................................................. 76
WebDAVS/SFTP/FTPS Access to Share Links ..................................................................... 76
Seafile ................................................................................................................................. 77
  Automatic Backup to ERDA ......................................................................................... 82
Scripted Server Backup to ERDA ........................................................................................ 102
  Schedule Tasks ............................................................................................................... 104
Data Import/Export ............................................................................................................. 107
Courses and External Collaboration .................................................................................... 115
  External Users / Peers ...................................................................................................... 115
  Personal Copy of Material / Import Sharelink ................................................................. 118


2 / 121
Introduction

ERDA or Electronic Research Data Archive at University of Copenhagen (KU/UCPH) is meant for storing, sharing, analyzing and archiving research data. The intended audience is employees, their collaboration partners and students. ERDA delivers safe central storage space for own and shared files, interactive analysis tools in addition to archiving for safe-keeping and publishing. You can use ERDA as a secure network drive from anywhere and it also comes with a stand-alone file synchronization service similar to Dropbox, but with the data safely stored locally at UCPH.

The latest version of this guide is always available from https://erda.ku.dk/ in different languages.

Requirements and Terms of Use

You only need a web browser to use the fundamental features of the ERDA system. In case you want to use the advanced features, you may have to install additional client software.

ERDA comes with few restrictions and rules, and you can always find them on https://erda.ku.dk/terms.html

In short the terms should follow common sense, e.g. it is your responsibility to only store data that you actually have the right to store, and to keep your login safe and secret to prevent others from access to your account.

IMPORTANT: Please carefully read about data replication and backup in our FAQ on the front page to make sure you are fully aware exactly what we do and don’t offer and what your own responsibilities are regarding data safety. This is relevant both in regard to our terms of use and fulfilling the requirements of the mandatory data classification at UCPH.

IMPORTANT: Please note that handling and storage of so-called personal data typically comes with strict regulation set out in The Act on Processing of Personal Data as described on GDPR and Research on KUnet and the information from the Danish Data Protection Agency. Thus you are not generally permitted to just store them on your PC or on remote storage solutions like ERDA. Please contact us about rules and requirements prior to storing any such data on ERDA, or use the sister system, SIF, which is officially approved for exactly that purpose.

How to Access UCPH ERDA

Everyone associated with UCPH has a username consisting of six characters, like abc123. The username is popularly known as the Swedish number-plate, and it’s generally used to access intranet facilities like KU Webmail KUnet.

You can use that same login to sign up for and access your account on ERDA.

Externals can in some cases also sign up as outlined in Courses and External Collaboration.

We also provide a 'digital signature' login method for users who prefer that solution or just don't have a KU login, but that is not covered in this guide. Please refer to our FAQ on https://erda.ku.dk for details.
**Sign-up**

**TIP:** We recommend following our new Sign Up Intro from [https://erda.ku.dk](https://erda.ku.dk) but include the steps here as well.

For KU/UCPH-employees and students signing up is just a matter of clicking **sign up** from the **KU / UCPH Users** tab on [https://erda.ku.dk](https://erda.ku.dk)

and entering your KU login when prompted.

As an example we show it for the user with KU username *brs278* below, but please use your own KU login.
When you click **Yes (Allow)** your account on ERDA is created and you will see a link to your personal entry page like this:
Automatic redirection should kick in but otherwise just click the your personal page link to go start using your account.
Login

After signing up you can always access your account by pointing a web browser to https://erda.dk and entering your KU login as shown below if prompted.

After entering your login and clicking Yes (Allow) you will reach the actual ERDA site.

The web browser remembers your login for a while, so you don't have to repeat the login every time. You can explicitly log out inside ERDA after use. This is generally recommended, and particularly if you share your computer with anyone else.

Overview

When you open ERDA you will start out on Home where you can access and manage your ERDA Apps. Use the navigation menu to the left to access to your Home, Files, Workgroups, Archives, Setup, Settings and Jupyter pages. Each of those pages is described more thoroughly in the following sections.
Home

On your Home page you can navigate between the available ERDA Apps and modify which apps to enable in your web interface.

Welcome to UCPH ERDA!

Tools from UCPH ERDA helps you with storage, sharing and archiving of data. UCPH ERDA delivers centralised storage space for personal and shared files.

Quick Tip: Easy data Analysis and Presentation ...

Your apps & app-setup

Click the Add tile with the white/green plus icon to add or remove apps from your personal pages.
Files

This is your main area on ERDA when it comes to data and it’s where you manage your personal and shared files. The directory tree in the left pane takes you quickly to a specific sub-directory and the right pane shows the files and directories in the current directory.

The location bar right above the two panes shows the path to the current directory with quick navigation links to parent directories.

Right-clicking a directory or file brings up a context menu in which you can select common actions on the item that you right-clicked. For files the actions include e.g. download, edit and rename. For directories you get the choice of e.g. uploading or creating files in-line. Right-clicking in the empty space below the entries in the right pane brings up the context menu for the current directory.
You can add new text files in-line with the Create File entry from the right-click menu or upload with the Upload File entry there.

The latter opens the upload dialog where you can drag and drop local files from your desktop or click Add files... and point files out for upload.

As an example you can select Create Folder, enter the folder name uploads and click OK. Then double click the new uploads folder to open it.
Inside the uploads folder you can again right click and now select **Upload File** to open the upload helper. Click **Add files…** to select the files you want to upload.
When you are done selecting files for upload you can click **Start upload** to actually upload.
wait for them to complete or click **Cancel** if you change your mind.
Finally click **Close** when finished and your uploaded files show up among your other files.
Double click files to open them directly in the browser if possible but with fall back to download and open in a suitable local application otherwise. In the latter case any changes will only be saved locally and not in ERDA. Therefore it is necessary to upload the modified file again in that case.

It is possible to edit simple text files with **Edit** from the right click and please refer to the sections about Advanced Data Access for a guide to the possibilities of mounting and easily working directly on files in your ERDA folder without worrying about the upload step.

**File Sharing and Data Exchange**

Research collaboration often requires some kind of data sharing/exchange. ERDA has a few convenient ways to provide that for common purposes. Explicit data exchange with the **Share Link** method is often the easy choice for one-time sharing or collaboration with external partners who are not ERDA users. For continuous transparent sharing purposes the **Workgroup Shared Folders** method is often more appropriate, because it automatically and dynamically shares the corresponding folder and contents with exactly the group of people maintained by the workgroup owner(s). More granular sharing is also possible using nested layers of workgroups.

**Share Links**

In ERDA you can use share links for quick and easy exchange of data. Just right click a file or folder on your Files page and select **Share Link** then **Create** beneath it. In this example the recently created uploads folder is chosen as the share link target.
Optionally tick the **Read Access** and **Write Access** check boxes in another combination to fit the kind of desired exchange and click **OK**. The three valid combinations result in read-only, write-only and read-write shares.
In this case it is the default read-only share, so that recipients of the share link can only view/download the files.

Once the share link is thus created you can **Open share link** to see how it will look to share link recipients.
You can manually distribute the URL of that page by any means you like, or you can use the built-in email invitations available through Edit and send share link from the previous share link dialog.
Enter one or more recipient addresses and optionally a message to send along with the link. Then click **Send Invitation(s)** to actually deliver the invitation.

If later you decide to invite more recipients or otherwise want to view, edit or remove a share link you can always use the Share Links app to do so. First navigate to **Home** and use the **Add** button with the plus to enable Share Links in the navigation menu. Then click the **Share Links** button in the navigation menu to the left to view and manage them.

Please refer to the WebDAV/SFTP/FTPS Access to Share Links section for further details about advanced uses of share links.

**Workgroup Shared Folders**

The other essential file sharing method for e.g. collaboration projects and research groups is workgroup shared folders. ERDA tightly integrates this workgroup sharing of files in the Files interface. For each workgroup you participate in there's a shared directory with the name of the group and a shared-folder icon 📀. All files and directories inside such directories are automatically and transparently shared with all participants in the group. In this example the user participates in the HPC/Docs group so the associated shared directory is shown. As seen it has a plain README file and two nested sub-workgroups, namely ERDA and SIF.
In the example above *Docs* and each of the *ERDA* and *SIF* subfolders are Workgroup shared folders nested under the *HPC* Workgroup, as indicated with the chain folder icons. I only requested membership of the HPC/Docs sub-workgroup here, so the parent HPC folder is marked as an ordinary folder.

More information about signing up for groups is available in the following sections.

**Workgroups (VGrids)**

You can share files with your colleagues using the Workgroup features.

**TIP:** We recommend following our new Workgroup Intro from [https://erda.ku.dk](https://erda.ku.dk) if you want to use Workgroups, but include the steps and further descriptions here as well.

On the Workgroups page you can see all such groups, and the ones you participate in have extra options like a link to the group shared files and web pages.
If you think you are entitled to participate in a group, that you are not already participating in, you can use the green request-membership icon to ask the group administrator(s) to give you access. The request-ownership icon can similarly be used to request co-administrator rights for a given group. When you click to request member or ownership a small dialog will open with a text field where you can write a message/reason, which the owners will receive along with your access request on email.

Privileged users can also add new groups using the Create Workgroup button at the bottom of their Workgroups page as shown in the example here.

Workgroup Web Pages

Workgroups come with a simple web hosting framework, which can be used e.g. to present your workgroup or project to the world at large and to the participants specifically.

When you create or become co-owner of a Workgroup, say XYZ, you automatically get two new directories in your ERDA home:

A) private_base/XYZ
B) public_base/XYZ

Any files you put in (A) are visible as web pages online but only to the Workgroup participants and any files you put in (B) are visible as web pages online for everyone. You can manage the web contents directly in those folders through your preferred ERDA file access method, or use the Edit
links in the Web Pages column on your Workgroups page for a simple online editor. The same location also has View links to the resulting web pages. It should be noted that the public web pages can be used without an ERDA account through the address:

https://erda.ku.dk/vgrid/XYZ

or on either of the aliases

https://erda.ku.dk/workgroup/XYZ
https://www.erda.dk/vgrid/XYZ
https://www.erda.dk/workgroup/XYZ

if case you have any preference for the naming used.

**Archives**

The Archives page provides a method to archive or freeze a snapshot of one or more files. This is required e.g. in relation to handing in PhD theses and in certain cases required or convenient when submitting a paper to a conference or journal. Initially your archives list is empty but after adding a few different archives it could look like the example below.

You can inspect a frozen archive with the green info icon, edit unfinished archives with the green wrench icon, and if the system is configured to allow deletion of archives you can remove them again with the red remove icon.

23 / 121
New archives can be created with the Create a new frozen archive link at the bottom of the page.

As an example we could create a new archive called Article Data ... by filling out the resulting form like this:

Existing private or shared ERDA files can be added with the Add file/directory button and new files can be uploaded directly to the archive with the Add upload button. The former button brings up a file select dialog in which you can select from your ERDA files: double-click individual files or right-click and choose select to pick entire folders. The latter button opens an upload dialog like the one from Files. When you are done adding files and have marked if you want the archive files to be published, you can click Save and Preview to inspect the current contents.
If you selected *Make Dataset Publicly available* you can use the **Preview publishing** button to see a draft of the published archive. At this point you can continue modifying the archive contents like above through the **Edit archive** button, until at last you click **Finalize archive** to actually permanently freeze it and thereby mark it ready for additional tape archiving. This is necessary to get the extra data safety and guarantee that it will remain available for at least 10 years.

Please note that archiving does **not** remove the files you may have selected from ERDA, but only saves a snapshot of the current contents. That is, you are free to keep working on such files afterwards and without causing changes to the archived version.
After finalizing you can use the **View details** button to see all the details registered about it on the view archive page and use the links to access the associated files.
For archives with publish enabled we also integrate access to request a Digital Object Identifier (DOI) after finalizing the archive. From view archive you click **Register Archive DOI** at the bottom to reach the central UCPH DOI registration portal. Typically this involves clicking through a standard UCPH login and small DOI intro dialog to get to the actual DOI metadata schema shown.
Once filled and submitted the request is sent through the UCPH validation procedure and if everything is okay you receive a permanent https://dx.doi.org/XYZ URL alias for your published archive data. From then on you can e.g. use it as a reference in research papers or provide it to research colleagues interested in re-using your published data.

**Jupyter**

ERDA integrates a set of Jupyter services, which can be used to easily perform a wide range of data analysis and visualization tasks directly on your ERDA data. We rely on the JupyterLab web interface to provide interactive and flexible *notebooks* with direct and efficient access to your ERDA home directory. In the notebooks you can use various popular programming languages like Python and R to explore, analyse or present your data, e.g. for your own research, for interactive presentations or even for teaching purposes.

To get access to these services, ERDA provides a Jupyter button in the navigation menu. Upon clicking it, the page to Select a Jupyter Service appears. On this page you are presented with a set of horizontal service tabs at the top, and each tab presents and describes the individual service and how it is configured in the Service Description. Below the description there’s a Start SERVICE button, which you can click to open a connection to that particular service in a new web browser tab or window.
like seen here after clicking **Start DAG**

By default, it will take you to your personal home page on the Jupyter service as shown above,
which is provided via our hosted version of JupyterHub. That is, the standard infrastructure to provide individual isolated Jupyter notebook instances to multiple users sharing a pool of actual compute nodes.

In other words it is mainly used to keep track of your running notebook instances, to let you transparently access them from wherever you may be and to enforce the resource and life-time policy for the individual instances. The latter is a matter of sensible resource utilization and in practice it a.o. means that disconnected DAG notebook instances will be terminated after two hours as stated in the service description.

You may notice that the JupyterHub service page comes in two flavors depending on whether you have or haven’t already spawned a notebook instance. If not you’re presented with a Start My Server button, that takes you on to select which notebook flavor to run. Otherwise the resulting page will instead let you open your current active notebook instance via the My Server button, or to shut it down via the Stop My Server button as shown here.

Upon clicking the Start My Server, the site will give you an option to choose which Notebook image you want to Spawn.
The dropdown menu here comes with a variety of images that will change over time as we introduce new ones or deprecate existing ones to each Service. The titles of each of these attempt to be fairly descriptive in terms of what purpose they imply to solve in terms of tasks that can be accomplished with them. For instance, the “Data science Notebook with Python” is a Python specific Notebook image that has an extensive number of modules and libraries that from experience and user requests are common in doing tasks related to data science. This includes python modules such as numpy, scipy, and pandas.

In the following picture you can see a snapshot of images we had available on DAG at the time writing.

Currently, every user has the ability to spawn each of these. The only exception to this is the DGX-1 Notebook and the GPU Notebook image, unless so granted an attempt to spawn these notebook images will fail and present you with the following message You don't have permission to launch that image. This is by design, since the DGX-1 notebook image gives access to a special purpose DGX-1 machine, dedicated to GPU intensive tasks by the Machine Learning group at DIKU. Therefore access permission must be explicitly granted by Erik Bjørnager Dam. Similarly the GPU notebook image is dedicated to a GPU host owned by the eScience group.

Upon spawning an available notebook image, you will be redirected straight to the JupyterLab interface as shown below.
For the **R Notebook** you can choose to launch **R Studio** from the JupyterLab top menu and for most of the other notebook images you have a selection of Python, R and C/C++ interactive notebook flavors to choose from in the Launcher window shown above. There’s also Console and Terminal buttons for those who prefer to work with low-level commands and scripting.

In the lefthand side of the interface, JupyterLab provides as per the documentation “a number of commonly-used tabs, this includes the file browser, the list of running kernels and terminals, the command palette, and a list of tabs in the main work area.”. The file browser on **DAG** by default contains the content of your ERDA home. An example of this can be seen below where...
both the ERDA File manager and the JupyterLab file browser are currently inside the personal ERDA directory /erda_folder.

Any files created inside this directory, via the JupyterLab file browser will be reflected into your ERDA home. For instance, if if create a new Notebook file (.ipynb) by clicking the Python 3 button in the Launcher window, by default a file called Untitled.ipynb will be created inside the
current file browser directory, which in this case is the /erda_folder. The name of the new file can be changed either by right clicking the file in the file browser, or by right clicking the new Notebook tab that appeared once the Python 3 button was clicked. An example of rename via this new tab can be seen below. Here the Untitled.ipynb file was renamed to be called my_first_notebook.ipynb

Since we selected that the new Notebook should use Python 3 in the previous step this selection is reflected in which kernel the Notebook will activate, in this case it will be Python 3 as can be seen from the “Python 3 O” logo at the righthand side of the Notebook's menu.

A kernel is a term for specifying how a particular Code Cell should be interpreted when it is run. A Cell is the rectangular input areas in which you enter text into a Jupyter Notebook via the JupyterLab interface. A highlighted example of an empty Cell can be seen below.
A Cell can contain any text that you input into it, however when a Cell is “Run”, either via the button or by the keyboard shortcut Shift-Enter, the Cell type decides how the contents should be interpreted. This is selected through the dropdown menu in the menu bar of the Notebook. By default it is set to Code, so that upon running the Cell the input will be interpreted by the active kernel which in this case means as Python 3 code.

As seen the print call gave the expected result by outputting the “Hello World” message. The active kernel and thus how the Code Cell should be interpreted can be changed via the mentioned “Python 3 O” logo. By clicking the “Python 3” part, a menu will appear as shown below, that allow you to select which kernel this active Notebook should use.
The options here are limited to the kernel flavors that the Notebook image supports, which means that they will vary from image to image. In “Datascience Notebook with Python” we support Python2 and Python3 kernels. As shown in the picture below, when switching to the Python2 kernel the logo is updated accordingly to “Python 2 O” and when a Cell is Run it’s interpreted with Python2 instead.
Beside the **Code** format, the **Cell** can be of two other types, i.e. **Markdown** or **Raw**. **Markdown**, for instance makes it possible to create HTML elements as shown in the picture below, where the 3rd **Cell** is an executed version of the 2nd. It can be used to format the Notebook as a textual structure with Headings, Sub-headings, Paragraphs, Lists etc. Checkout [https://guides.github.com/features/mastering-markdown/](https://guides.github.com/features/mastering-markdown/) for examples on how to do this.
Finally **RAW** does what it says, it doesn't try to interpret the text but simply outputs it as is.

When a **Cell** is being run by the **kernel**, the JupyterLab interface will indicate this by updating the empty “[ ]” next to the **Cell** to a “[ * ]” as shown in the picture below. This means that the **kernel** is not finished running the **Cell**, which could either be because it is waiting for previous **Cell**s to finish first or that the tasks just takes a long time to finish. If for some reason the **Cell**'s indicator changes state from “[ * ]” with no further activity “[ ]”, it is likely that a **Cell** has failed to run and the task was killed mid run. E.g. because the notebook reached the limit on maximum allowed memory use. We’ve added a little memory-meter top right as shown with 71MB used below to help detect such problems.
Upon a successful run, the * inside “[ * ]” will change to a number that is equal to how many Cell's that were run before it and add 1 to it as shown in the page below.
Finally the **Logout** button is designed to end your authenticated session on a typical JupyterHub site, but since we transparently authenticate you via the ERDA service, clicking this button has no effect on your connection to the service.

For additional up-to-date information on the JupyterLab interface itself please see the [official documentation site](#).

### Jupyter Notebook Customization

The provided notebook images offer a wide selection of software libraries for specific research areas, but in case you are still missing a particular software or library you might still be able to install it yourself or have us add it in general. Please note that commercial/closed-source software is usually difficult or impossible to integrate due to various licensing issues when used in a data center.

### Python

For Python and Jupyter extension packages it may work to simply install temporarily with pip or pip3 inside the session via the **Terminal** in the main Jupyter Launcher window, or by writing a standard Python `requirements.txt` file in your ERDA home and run it interactively like `pip3 install -r requirements.txt` from the Jupyter Terminal.

Please note that you need to use `python2` or `python3` specifically if you want to use such `pip` or `pip3` installed packages respectively directly from the Terminal.
Alternatively you can install Python software dependencies like e.g. the *scikit-image* package directly in the Jupyter Notebook itself by evaluating

```bash
!pip3 install scikit-image
```
in a cell.

At the moment installing software that requires compilation or come with other external dependencies is not generally supported.

To ensure that packages you install will stick around between different sessions, pass the `--user` flag to the `pip` or `pip3` install command. This will ensure that the package is installed into your own home `{service}_config_` directory, which persists between the individual Jupyter sessions.

**R / RStudio**

For R packages, DAG supports personal persistent installation into your own workspace. When utilizing the `install.packages()` command in the Terminal, Notebook, or R-Studio the designated packages will be installed into your local `~/work/__dag_config__/R/libs/` directory, which is hosted on ERDA and thus preserved across DAG sessions.

Subsequent to updates being released to DAG notebooks, issues might be encountered with version conflicts between your old persistent packages and the updates. The most straightforward solution then is to clear out your old packages and install a new version of them. This can be done either through the Terminal with

```bash
rm -fr ~/work/__dag_config__/R/libs/*
```
or in an R environment with:

```r
unlink("~/work/__dag_config/R/libs/*", recursive=TRUE)
```

Afterwards the packages can be installed with the mentioned `install.packages()` command if needed.

**Other Packages and Install**

It is also possible to use the *conda* package manager to install additional software in running sessions. You can use

```bash
conda install -y -n ENV PACKAGE
```
from the Jupyter Terminal where ENV is typically *python2*, *python3* or *r* depending on the notebook and your working environment. Similarly use

```bash
conda search NAME
```
to search for available packages.

**MATLAB programs on DAG and MODI**

In order to run MATLAB programs on DAG/MODI, you’ll need to compile your program to a standalone application without display. It must be built to run on the current Ubuntu LTS, which is at the time of writing version 20.04. If you don’t already have Ubuntu 20.04 as default, you can use a VM with Ubuntu 20.04. Afterwards, you can install MATLAB and its compiler from MathWorks, which UCPH users have a site license to do.

We have tested the guide with success on a native (64-bit) Ubuntu 20.04 machine and on a Ubuntu 20.04 Virtualbox image. The latter image can be downloaded from [https://www.osboxes.org/ubuntu/](https://www.osboxes.org/ubuntu/)
First you make any adjustments or wrapping of your MATLAB program, to allow it to run as a headless script on DAG and/or MODI. That part is outside the scope of this guide but you can find help from MathWorks or online searching e.g. for *MATLAB headless nodisplay*.

You then enter MATLAB, and do as shown in the following example:

You click the “Apps” tab.

![Apps tab](image)

Choose “Application Compiler”

![Application Compiler tab](image)

Add your file, and make sure to check “Runtime included in package”.

![Application Compiler settings](image)

Go to settings and add the following parameters. Press “Apply” and “OK”.

42/121
Press “Package” and wait for the compilation. It might take some time.

Once your compilation and packaging is complete, you’ll only really need the resulting installer found in the folder “for_redistribution”, to run your application.

For easier implementation it is recommended to create some custom shell scripts, which you add to your upload folder. To get a feel about how this will look like, the next part of the guide will consider implementation of our simple “helloworld” example. You can find the location of the example installation folder at [https://sid.erda.dk/sharelink/b3HZc5HEqa](https://sid.erda.dk/sharelink/b3HZc5HEqa)

**Helloworld example**

Start out with the common steps. If you perform them in ~/erda_mount on MODI or in ~/work on DAG, you don’t have to redo them every time you want to run the program, because the files will be stored on ERDA and exposed through the default network drive mapping.

**Common**
1. Start the terminal
2. Navigate to your desired installation folder. The root folder is used here.
3. wget [https://sid.erda.dk/share_redirect/b3HZc5HEqa/helloworld.zip](https://sid.erda.dk/share_redirect/b3HZc5HEqa/helloworld.zip)
4. unzip helloworld.zip
5. cd helloworld
6. chmod +x *.sh

Which looks like the following in the Jupyter Terminal
```
jovyan@b00a9aaea96e5:~$ wget [https://sid.erda.dk/share_redirect/b3HZc5HEqa/helloworld.zip](https://sid.erda.dk/share_redirect/b3HZc5HEqa/helloworld.zip)
jovyan@b00a9aaea96e5:~$ unzip helloworld.zip
jovyan@b00a9aaea96e5:~$ cd helloworld
jovyan@b00a9aaea96e5:~$ helloworld$ chmod +x *.sh
```
After you’ve followed the common steps, you either run the DAG steps or the MODI steps based on which service you intend to use.

**DAG:**
- Locate helloworld folder
  - `~/helloworld$ ./install.dag.sh`
  - `~/helloworld$ ./helloworld.dag.sh`

**MODI**
- Locate helloworld folder
  - `~/helloworld$ ./install.modi.sh`
    Run on frontend
  - `~/helloworld$ ./helloworld.modi.sh`
    Run on node
  - `~/helloworld$ ./helloworld.modi.slurm.sh`

In “~/helloworld” only the installer is generated from MATLAB. All the other files are shell files that give the installation of your program more structure. Even though they are designed for this specific example, a lot of the code is still reusable. Therefore, it is recommended to study them, so you can create your own. In this example they were included in the same folder as the installer.

**Settings + Setup**

On the **Settings** and **Setup** pages you can tweak your personal ERDA settings to your liking. This includes configuring the look-and-feel of the web interface and setting up advanced access methods like e.g. access with a SFTP/WebDAVS/FTPS/Seafile client or mapping of your user home as a network drive on your local PC or workstation.

ERDA also includes a file synchronization service like Dropbox, but one which KU employees are permitted to use – unlike Dropbox.

Further description of the **SFTP/WebDAVS/FTPS/Seafile** tabs and their use is available in the Advanced Access sections.

You don't need to edit the settings to just use the fundamental functionalities. So you can safely leave them alone if in doubt.
If you mostly use a particular page like e.g. Files on ERDA web, you can change Default Page to that in order to always launch it directly on login.

Changing Site Collaboration Links from default to advanced enables additional features on your Workgroups page.
Setup provides access to a number of slightly more advanced ERDA features, which we will inspect closer in the second part of the user guide.

2-Factor Authentication

Recently there’s been a general rise in automated password guessing attacks and phishing attempts to obtain login information from users on the Internet. So it is a good idea to take extra precautions about securing your login credentials against illegal access and abuse.

All ERDA services offer optional 2-Factor Authentication (2FA) as a means to let you do just that. In practice 2FA adds an extra step to your logins where you prove that you don’t only know something – namely your username and password – but that you also possess something nobody else does. In ERDA the latter is in the form of a simple personal one-time token generator. So after entering your usual login you are asked to enter the current token to prove that it is really you and not somebody who somehow managed to get hold of your password.

ERDA uses the TOTP standard for the one-time tokens and you can use an app on your smart phone or tablet as the token generator.

Open the 2-Factor Auth tab on your Setup page and follow the four-step wizard there to prepare your device and enable this extra security measurement.

Start the setup wizard by clicking **Okay, let’s go** and install a TOTP authenticator app on your phone or tablet using your usual app store. We’ve mainly tested with Google Authenticator and FreeOTP but any of the four listed plus a number of other TOTP authenticator apps should work.
Then click I’ve got it installed to proceed.
You can now import your personal 2FA secret either by scanning a QR code or by typing the raw text code. The former is easier if your device has a built-in camera. Open the previously installed authenticator app and import the QR code e.g. by choosing Scan a QR code as shown for the Google Authenticator.

Please note that the exact method varies depending on the platform and app of choice.
Click **QR Code** in the ERDA wizard to view your personal QR code and then point the device camera towards it to scan it.

The apps typically react by adding a new entry displaying the current token (a number) once the scan succeeds as shown below.
With your 2FA setup thus in your app you click **Done importing** and enter the 6-digit token currently shown in the app in the resulting token popup.

In this case the app showed the token 734572 and it was therefore typed in before confirming with **Verify**. Please note that the tokens are short lived and a new one is generated every 30 seconds. You can just try again if fails because it changed in between typing it and verifying.

Once accepted the wizard proceeds to the last step where you actually enable 2FA for one or more purposes.
your phone or tablet. You can find and install either of them on your device through your usual app store.

2. Import Secret in Authenticator App
Open the chosen authenticator app and import your personal 2-factor secret in one of two ways:
A. Scan your personal QR code
B. Type your personal key code

The latter is usually more cumbersome but may be needed if your app or smart device doesn’t support scanning QR codes. Most apps automatically add service and account info on QR code scanning, but otherwise you can manually enter it.

3. Verify the Authenticator App Setup
Please verify that your authenticator app displays correct new tokens every 30 seconds before you actually enable 2-factor authentication. Otherwise you could end up locking yourself out once you enable 2-factor authentication.

4. Enable 2-Factor Authentication
Now that you’ve followed the required steps to prepare and verify your authenticator app, you just need to enable it for login below.
This ensures that your future UCPH ERDA logins are security-enhanced with a request for your current token from your authenticator app.
SECURITY NOTE: please immediately contact the UCPH ERDA admins to reset your secret 2-factor authentication key if you ever lose a device with it installed or otherwise suspect someone may have gained access to it.

Enable 2-Factor Authentication
Add an extra layer of security to your UCPH ERDA login by requesting a new token on your phone or tablet.

Enable 2-Factor Authentication
Log in to your UCPH ERDA login with your username and password, and then click the Sign Out button from the user menu in the lower left corner and click Yes to actually end your current login session. This is of course particularly important if you share your computer or device with anyone else.

Finally remember to Save 2-Factor Auth Settings in order to actually enable it, so that on future logins you will be prompted for the current 2FA token after the plain login with username and password.
Advanced Data Access

In addition to the web interface ERDA also provides some advanced access interfaces for efficient and transparently integrated file access. These interfaces e.g. allow you to transfer many or big files more efficiently and to map your ERDA files and folders on your local PC or workstation like a network drive. In that way you can work with them remotely as if they were local files. All the interfaces use a secure connection to ERDA, so they are accessible from anywhere where you have an Internet connection. Thus, you do not need VPN or anything like that to access them from outside UCPH.

**TIP:** We recommend following our new Network Drive Intro from [https://erda.ku.dk](https://erda.ku.dk) if you want to use ERDA as a network drive on Windows, but include the steps for all platforms here as well.

ERDA also offers a file synchronization service resembling Dropbox, but one that employees/students at KU are permitted to use in contrast to Dropbox.

**TIP:** We recommend following our new Seafile Intro from [https://erda.ku.dk](https://erda.ku.dk) if you want to use ERDA Seafile, but include the steps further down as well.

**WebDAVS**

WebDAV is a popular protocol to access remote storage using extensions to the HTTP protocol, and it can be secured with TLS/SSL in the same way. In the following sections we refer to this secured version as WebDAVS. Recent versions of Microsoft Windows, Mac OSX and Linux support this protocol out-of-the-box, so with those it's possible to access such remote content like it was in a local folder. That is, without the need to install additional software and still in a secure fashion. For that reason it is relatively straightforward to get started with working on your data through WebDAVS. However, it should be noted that WebDAVS was built for simple web page publishing, so it is neither particularly efficient for bigger data transfers nor robust against network glitches. Thus, we generally recommend looking at one of the SFTP-based solutions, especially if performance and stability is a concern.

You generally need to first open the **WebDAVS** tab on your **Setup** page at ERDA and configure how you want to authenticate to our WebDAVS server. Enter a password that you want to login with and click **Save WebDAVS Settings**. Please also note your Login Details including your automatic login username shown there.
You can also find the required server URL by clicking the Show more WebDAVS client details... link there. In this example the server URL is https://io.erda.dk and the automatic username is bardino@nbi.ku.dk, but please look up your own.

**Windows File Manager**

In Windows it is possible to access WebDAVS locations directly from the built-in file manager. E.g. with the Map network drive or the Add a network location button available when navigating to the This PC location as shown in the images below. Please note that the screens vary slightly with different Windows versions and it is also possible to use other commercial solutions than the built-in File handler.

Below the procedure with the native Windows 10 File handler with Map network drive is shown.
Click **Computer** in the top menu to expose the **Map network drive** button
which is clicked to open the connect network drive wizard and connect to ERDA.

Enter server URL from the WebDAVS tab – in this case https://io.erda.dk – then click Finish.
You will then be prompted for your automatic username and the password you entered on the WebDAVS setup tab as described on page 52. In this case it specified the automatic username bardino@nbi.ku.dk.

In case of correct log in ERDA is attached as network drive with the chosen drive letter (Z: here) and the drive is opened in a new window. That is, your files and folders on ERDA appear in the usual file handling.
As long as you are online you can use the files there just like your local ones. Next time you start your computer you can easily access the files again using the saved ERDA network drive from the same location as shown in the first screenshot in this section.

You can detach the network drive again by locating it as above (Z: here), right clicking and choosing Disconnect.
then the drive is detached and the icon disappears.

**Adjust Windows WebDAVS Transfer Limit**

Unfortunately, Microsoft decided to enforce a very low, artificial limit on the size of files that can be transferred to or from WebDAVS locations.

In practice, the limit implies that the user receives a rather uninformative error message e.g. if files larger than about 50MB are attempted downloaded from the WebDAVS folder. Example shown below.
It is possible to increase this limit to 4GB by applying some modifications in the **Registry Editor**.

This can be done by replicating the following steps.

First, search for `regedit` and run the command.
Afterwards, open the folder **HKEY_LOCAL_MACHINE**.

Open **SYSTEM**.
Open **CurrentControlSet**.

Open **Services**.
Open WebClient.

Open Parameters.
Set the value in **FileSizeLimitInBytes** to 4294967295 in Decimal base.

ERDA WebDAVS network drive now allows upload/download of files up to 4GB.

In general, we still recommend to use SSHFS over WebDAVS, especially if you work on large amounts of data.

**Mac OSX File Manager**

On Mac OSX you can access remote WebDAV(S) content with the built-in file manager, Finder. In the menu under **Go** you select **Connect to Server**.

Enter the server URL from the **WebDAVS** setup tab and click **Connect**.
You will then be prompted for your automatic username and the password you entered in the WebDAVS tab as described on page 52. In this case it specified the automatic username bardino@nbi.ku.dk.

On success you will see the remote contents which could look like this

65 / 121
Linux File Manager

On most modern Linux distributions you can access remote WebDAV(S) content with the native file manager - be it Nautilus, Konqueror, Dolphin, Thunar or something else. In the file manager menu typically under Go you select Open Location.

Enter the server URL from the WebDAVS setup tab (possibly replacing https with davs in the URL if it complains) and click Connect. You find the server URL.

In XFCE with the Thunar file manager as shown here you need to use the davs rather than the https prefix in the URL.
You will then be prompted for your automatic username and the password you entered on the WebDAVS setup tab as described on page 52. In this case it specified the automatic username bardino@nbi.ku.dk.

On success you will see the remote contents which could look like this
NOTE: The Nautilus/Thunar file managers rely on the GNOME gvfs library, which had a known bug at least until recent versions. Thus upload of files bigger than 1GB would simply hang. You can try to upgrade to a working version of gvfs (e.g. 1.22.2) or use another client if you experience this problem.

**SFTP**

SFTP is a secure and efficient file transfer protocol similar to the old FTP protocol, but with proper security built in. It relies on the security mechanisms of the OpenSSH suite of applications and it is supported by a number of clients on all popular platforms. Some platforms even integrate it natively in the general file manager, so that no extra software is required.

You generally need to first open the SFTP tab on your Setup page at ERDA and configure how you want to authenticate to our SFTP server. Either by inserting one or more SSH public keys\(^2\) or entering a password that you want to login with and click Save SFTP Settings. Please also note the Login Details including the automatic username shown there.

---

\(^2\) How they are used is outside the scope of this guide but a good starting point could be [https://support.cs.ncl.ac.uk/Advice/Secure_Shell/PublicKey/](https://support.cs.ncl.ac.uk/Advice/Secure_Shell/PublicKey/)
We recommend one of the following clients known to work with ERDA but it is possibly to use others.

**FileZilla**

FileZilla is a stand-alone FTP/FTPS/SFTP client available on all popular platforms. It includes a **synchronized browsing** feature, convenient for manually synchronizing folders between your computer(s) and ERDA. Additionally FileZilla comes in a **portable** version, which runs from anywhere without the need to install it first - and thus no need for administrator privileges:


Open the **Site Manager** e.g. from the **File** menu in FileZilla and add a New Site with the ERDA details filled in as shown below, but just using the automatic username and the password you entered on your SFTP Setup tab:
First time you connect you are asked to verify the ERDA server *fingerprint*, which you can find on your ERDA SFTP Setup page tab as described on page 68. If you don't want to repeat this every time, you can tick the *Always trust certificate in future sessions* box before you choose OK to connect.
Then your ERDA files should appear on the right side pane like this
WinSCP

WinSCP is another popular stand-alone FTP/FTPS/SFTP client available only for Windows. The ERDA configuration is not that different from the FileZilla one, and likewise it comes in a portable version which doesn't require installation and administrator rights. Additionally it has a semi-automatic sync feature which eases synchronization of files and folders between your computer(s) and ERDA.

Please refer to the project page for more information:
https://winscp.net/

PuTTY SFTP

PuTTY is a popular stand-alone SSH client originally developed for Windows, but now also available on other platforms. It is used under the hood for a number of other clients and comes with a SFTP command line client that can be used to interact with ERDA.

IMPORTANT: due to security concerns you need a recent version to connect to our SFTP service. Version 0.67 and later are known to work, whereas older versions at least up to 0.64 fail to connect with the error “Couldn't agree a key exchange algorithm: available ...”. If you look at the PuTTY
Changes page you'll see that there are a number of recent security fixes, which motivate upgrading to version 0.68 or later anyway. Please refer to the project page for more information:

www.chiark.greenend.org.uk/~sgtatham/putty/

Linux File Manager

On most modern Linux distributions you can access remote SFTP content with the native file manager. The steps are the same as described in the corresponding section about Linux File Manager for WebDAVS, just with the URL changed to point at the SFTP service.

Enter the server details from the SFTP setup tab (possibly replacing sftp with ssh if it complains) and click Connect. In this example it would be just sftp://io.erda.dk as port 22 is implicit.

![Open Location](image)

Then proceed as for the WebDAVS case, but again with the login details from your SFTP Setup tab as described on page 68.

SSHFS

On Linux and Mac OSX it has already for a long time been possible to map/mount remote storage with SSHFS as a kind of network drive, so that the remote contents look just like if they were in a local directory. Recently a fully functional and client for Windows gained momentum to finally complete SSHFS support on all three major platforms. SSHFS uses SSH/SFTP connections underneath, so it inherits a number of the advantages like efficiency plus network robustness from there, and it works directly with ERDA SFTP.

You can install SSHFS on Windows by downloading and running WinFsp and SSHFS-Win from Win-SSHFS. That extends the usual map drive functionality in the file manager (Windows Explorer) to support SFTP like it does natively for WebDAVS. That is, similarly to the description in the Windows File Manager section for WebDAVS, but without the cumbersome extra system tweaks needed for WebDAVS.

There's a little video on the aforementioned Win-SSHFS page briefly showing a practical example. As it shows you can use the format `\sshfs\USERNAME@SERVER` in map drive, where USERNAME and SERVER are as described in SFTP above, and you are then prompted for your chosen password during login.

On Mac OSX you can install SSHFS by downloading and opening FUSE For OS X and SSHFS from OSXFuse SSHFS or via MacPorts/Brew commandoes. With the former you can additionally add MacFusion for a pretty graphical wrapping of SSHFS instead of using terminal commands.
On Linux you can install SSHFS in your Software Center or through the APT/YUM commands, where the package is simply known as sshfs. In addition to the usage example in your SFTP tab on Setup at ERDA, there are a number of tutorials online, if it’s not trivial to get it working. You can of course also just choose to rely solely on the built-in SFTP support from Linux File Manager, if you don’t feel like installing sshfs and figuring out how to use it from the command line.

**FTPS**

FTPS is another secure and efficient file transfer protocol built on top of the old FTP protocol, but with proper security added. It relies on the security mechanisms of TLS/SSL and it is supported by a number of clients on all popular platforms. As far as we know none of the major platforms integrate it natively in the general file manager, so you probably need to install a dedicated client to use it.

You generally need to first open the FTPS tab on your Setup page at ERDA and configure how you want to authenticate to our FTPS server. Enter a password that you want to login with and click Save FTPS Settings. Please also note the login details including the automatic username shown there.

Our SFTP server is more thoroughly tested and allows the stronger public key authentication, but our FTPS server may deliver higher throughput, so feel free to use either in accordance with your needs.

You can use the clients that we mentioned in the SFTP section, but other FTPS clients may also work.
FileZilla

FileZilla can be used to interact with FTPS on ERDA. It is done much like for SFTP, so please refer to the FileZilla SFTP section for more details.

In FileZilla open the Site Manager e.g. from the File menu and create a New Site with the ERDA details as shown below, but just using the automatic username and the password you entered on your FTPS Setup tab as described on page 74:

First time you connect you are asked to verify the ERDA server fingerprint, which you can find on your ERDA FTPS Setup page. If you don't want to repeat this every time you can tick the Always trust certificate in future sessions box before you choose OK to connect. Please note that the fingerprint changes occasionally so you may need to repeat this step some months later.
Then your ERDA files should appear on the right side pane as for the SFTP case.

**WinSCP**

WinSCP can also be used for FTPS. Please refer to the WinSCP SFTP section for more details.

**WebDAVS/SFTP/FTPS Access to Share Links**

It is possible to combine the ERDA Share Links with the efficient file access from WebDAVS, SFTP and FTPS. Thus share links created with read and write access are additionally accessible through these protocols, and can be locally mapped as a network drive. In that way even collaboration partners outside ERDA can use efficient file access to exchange data with you.

Similarly a share link on a personal/workgroup folder can be used to share it as a kind of common network drive on e.g. lab-PCs and such, where you don't want to leave your own ERDA account logged in.

Please refer to the details about mapping from different platforms in the previous sections and just note that instead of supplying your personal credentials you have to provide the 10 character share link ID as username and password. As an example you can access the share link with ID gBf51LXf4T by connecting with WebDAVS/SFTP/FTPS as previously described, but enter gBf51LXf4T as username and as password.
**Seafile**

Seafile on ERDA is a file synchronization service much like Dropbox – but since ERDA is located at UCPH physically and network-wise it comes without the UCPH restrictions on the use of Dropbox only for public data. All data are sent through a secure connection to the ERDA storage, and users can additionally choose to have their synchronized data encrypted locally before ever sending it over the network to the storage. This means that data are significantly better protected against eavesdropping by strangers. The optional encryption is done with the AES 256 algorithm, which 'Datatilsynset' and other authorities define as a suitable algorithm for strong encryption.

Seafile provides sharing capabilities which can be used to share data with internals and externals, and in that context it is left to the users to adhere to rules and regulations about confidentiality and sensitive private data.

You generally need to first open the **Seafile** tab on your **Setup** page at ERDA and register your Seafile account on ERDA. Enter a password that you want to use for Seafile login and click **Register**.

The registration will proceed and inform the ERDA administrators to activate your Seafile account.
Registration complete, please wait for administrator to activate your account.

You'll receive an email once your account is activated, and then you can use the Login link there to login with your email and the password you chose in the previous register step as described on page 77. Then proceed creating libraries of files and folders for synchronization and/or sharing.
As an example you create a library called autosync by clicking the New library button shown above and filling the pop-up with just that name before clicking Submit.

Then you can click on the newly created autosync library and upload files using the Upload button and selecting one or more local files as shown with the instructions.pdf file below.
Once you have data in your Seafile library you can share them with others by moving the mouse over the file or folder and clicking the **share** icon appearing.
This will open a short sharing wizard pop-up where you first Generate a sharing link and then enter email addresses of people to share it with. You can password protect the share and include a small message with the link email if you like.

To actually synchronize one or more libraries to your PC(s) or workstation(s), you need to install the client. When you login to your Seafile page there's a link to the downloads page at the bottom. The client is available for all common platforms, including Windows, Mac OSX and Linux in addition to clients in the app stores for mobile devices.

Download and install the one for your platform to proceed. Then add your account as shown here but with your email and password from the Seafile registration.
When you click **Login** you will get a list of libraries for your account. You can right click on a library and select **Sync this library** to make a local version for synchronization. You will then be asked where to store the synchronized folder locally. Once chosen the client should start actually synchronizing your library to that folder.

Any future changes or additions you make to that folder will also automatically be synchronized to your library.

Seafile also includes automatic file revisions, so that you can always fetch older versions of synchronized files on your Seafile web page. This includes files that you deleted from your local synchronization folder, making it work as a backup as well.

You can install the client on as many devices as you want and repeat the **Sync this library** to always have files synchronized between them.

**Automatic Backup to ERDA**

It is possible to combine Advanced Data Access with the free and open backup program [Duplicati](https://duplicati.org) to achieve automatic backup of your computer(s) to ERDA. Duplicati stores backup files in a special container form, which allows it to store e.g. special files and exotic file names, that may otherwise be a problem to save directly on ERDA. It also means that it's easy to encrypt the entire backup, so that no one else can read it, even if they should illegitimately obtain access to the ERDA storage.

Duplicati exists in a user-friendly graphical flavor as well as a command-line flavor better suited for backup of e.g. servers and certain NAS-systems. In this guide we only cover the graphical flavor.

In short setting up Duplicati requires two steps, namely creating the ERDA-specific configuration at
your ERDA Duplicati Settings followed by import of the resulting configuration(s) in the Duplicati client on your computer.

First you go to your ERDA **Setup** page and choose the Duplicati tab.
Enter one or more names you want to use for your backup(s) in the Backups field. E.g. the name of the computer or the folders you want to backup.

The remaining fields are optional and can just be left as is. Please note that you need to have configured password login for SFTP or whichever Protocol you choose for Duplicati, however. How to set up password login for the individual protocols is covered in the previous sections. Click **Save Duplicati Settings** when done and wait for the automatic page refresh.
where links to a .json file now appear for each backup name you entered in the Backups field. In this case we just entered Documents and got Documents.json as a result. You can download the json file(s) by right clicking on the link(s) and then something like Save link as or Save target as depending on which web browser and language you use.

The second half of the setup is to import the resulting json file(s) directly in Duplicati.

Download and install the latest 2.x release from

https://www.duplicati.com/download

Prepackaged versions for a wide range of platforms are available there. So download and and install the one which fits your computer. In this example we used the file with the -x86.msi suffix since it is suitable for 32-bit Windows systems like the one used here. For 64-bit Windows one would instead use the file with the -x64.msi suffix, and for Mac it would be the .dmg file.
On most platforms installing is a matter of double clicking the downloaded file and clicking through the install wizard. For Windows platforms the Windows Update utility should already have installed the required .Net 4.5+ libraries. On other platforms it may be necessary to install the corresponding Mono 3.0+ libraries before the installation will proceed.\(^3\)

When installed Duplicati should automatically start and show the status overview. However, you can always open Duplicati by right-clicking the status icon in the system tray next to the clock and choose Open there.

Select **Add backup** from the menu in Duplicati

---

\(^3\) If you run into problems with Mono and security warnings, you may also want to have a look at https://github.com/duplicati/duplicati/wiki/SSL-TLS-support-in-Mono
and choose the Import from a file option before clicking Next.
Here you **Browse** to the saved .json file in the Configuration file field and click **Import**.

Optionally set up encryption for the backup before clicking Next. Here we show it without encryption. It is left for you to weigh the need for encryption against the extra load it adds on your computer when encrypting and decrypting backup data. In case you backup confidential data it is definitely a good idea to encrypt it – possibly even a legal requirement in line with the guidelines from the [Danish Data Protection Agency](https://www.dsf.dk/).
On the next page all ERDA settings are pre-filled from the configuration file – except the password field. This password is the one you previously configured for the selected protocol on ERDA. I.e. with the default choices, it is the SFTP password you set on your ERDA SFTP Setup page, and so on.

At this point you can click **Test connection** to verify that the connection and login is correctly configured before proceeding with Next.
Select which local folders to include in the backup, like e.g. *My Documents* as shown here and click Next.

Please note that by default Duplicati runs on behalf of your regular user, so it may not have access to all system files or to shadow copy facilities available on some platforms. In effect this means that you may see warnings or errors if you select folders outside your user space or e.g. folders with application data in active use. Either leave out any such folders from your backup selection, or follow the Duplicati documentation on running Duplicati as a privileged user.
Optionally further tune the frequency for automatic backup runs and click Next. You don't need to touch the **Allow days** - no specific days selected has the same effect as ticking them all.
Optionally change **Keep backup** to e.g. 3 years or something and click **Save**.
If you didn't choose encryption in the earlier step you will be asked if you are sure you don't want that. Either answer **Continue without encryption** to proceed or **Cancel** to go back and set up encryption.

Now backup is ready to proceed and you can click **Run now**, etc.
When the backup run is complete you should see a status like below.
Repeat the **Add backup** procedure in Duplicati for any other backup .json files you created in ERDA Duplicati Setup.

After a successful backup run you can always recover selected files or folders from backup with the **Restore** entry in Duplicati or by folding out the backup entry and selecting **Restore files**...
Duplicati will then guide you through the process of restoring the desired data.

In case you lost your computer or got a new one you have to first download the .json files and import them to recreate the Duplicati setup.
Select the re-downloaded .json file
Again fill in your chosen password from the corresponding Setup page on ERDA. In this example it is the password from WebDAVS Setup.
Just click **Connect** unless you have encrypted your backup. Otherwise first enter the passphrase and change `--encryption-module=null` to `--encryption-module=aes` under **Advanced options.**
Now the backup configuration is fetched from ERDA.
Finally choose where the configuration should be saved and click **Restore**.
Then you can restore the actual data from your backup(s) on ERDA.

Further information on using Duplicati is available at https://www.duplicati.com/articles/Getting-Started/

**Scripted Server Backup to ERDA**

Some research groups have their own NAS / server for group data. If you want to backup the data from such a server to ERDA, but for some reason can't or don't want to use Duplicati, there are still a couple of alternatives for making scripted backup archives.

In short the simplest way of setting up something like that involves a Workgroup, upload through a read+write share link and a Workgroup workflow rule to trigger backup archive creation when a special file marker is uploaded.

First either choose an existing Workgroup of yours or create a new one for the backup as described in Workgroups (VGrids). In this example we'll use our existing Bardino Workgroup, but just choose a name that suits your setup. Now we create a sub-folder, say backup, inside the corresponding Bardino shared folder. For security reasons you probably don't want to save your ERDA password on the host to backup. To avoid that we can either use sftp/sshfs with an ssh key or instead make a share link with read and write access on Bardino/backup folder as described in Share Links. The latter allows us to access that folder using WebDAVS/SFTP/FTPS Access to Share Links, i.e. with just the share link ID as login and password instead of your general login. Access is then also limited to that particular sub-folder, which limits the risk of access abuse.

From the storage server we can then use a command line client like sftp or lftp to upload files there.
or even mount it with e.g. `sshfs` and use something like `rsync` to only upload file changes.

From that point we *could* of course open the ERDA web pages in a browser and manually create archives, but for most server administrators this step is something to rather include in the backup process from the server. So instead we set up a Workgroup workflow rule on the *Bardino* workgroup to enable just that. First go to your ERDA Setup page and enable access to the workflows by selecting *advanced* rather than *default* in the *Site Collaboration Links* drop-down. Now when we open the Workflows page it will include a *Workflows* column. There we click the *Open* link for the *Bardino* Workgroup to get to the Workflows page. We fill in the form as shown below to create a new workflow rule.

![Workflow form](image)

In effect it creates a backup archive of the sub-folder X whenever we upload/write a file called `X.create-backup` inside our *Bardino/backup/* folder.

Thus, when we upload a snapshot of the data from the storage server to *Bardino/backup/snapshot-28052017* and afterwards upload any file to *Bardino/backup/snapshot-28052017.create-backup* it results in the automatic creation of a backup archive called `snapshot-28052017` as the following workflow log displays.
Of course the resulting backup archive then also appears on the Archives page.

Alternatively to the workflows method you can use Schedule Tasks on ERDA to create backup archives of a folder e.g. every night. Please refer to the Schedule Tasks section for further details.

**Schedule Tasks**

It is possible to have ERDA do a number of tasks on behalf of you with the Schedule Tasks functionality. The corresponding navigation menu entry is not shown by default to keep it simple, but you can Add it from your Home page as shown. Click Save and wait for the for the refresh to let the Schedule Tasks button show up in the navigation menu to the left.
On the **Schedule Tasks** page you can schedule one-time or regularly repeating tasks
In the above example we have inserted a task to create a new backup archive every day at 16:17 with the contents of the uploads folder on ERDA. One can then either manually have saved data in that folder or used methods like outlined in the Scripted Server Backup to ERDA section to regularly upload data from other sources there. It should be noted that the createbackup command uses an archive name with three special automatic variables like +SCHEDDAY+ etc. to label the archive with the task run date.

You can follow the scheduled runs in the View Logs tab and the log from the first run of the above looked like the following:
As the log shows the backup archive has automatically been given a name to match the current date. You can read more about the automatic variables and available commands at the bottom of the Manage Tasks tab on the Schedule Tasks page.

**Data Import/Export**

In case you have other data storage servers or instruments on the Internet, it may be useful to be able to import data to – or export data from ERDA in an efficient way. Typical scenarios include fetching big data sets from dedicated Linux/UNIX servers or from e.g. imaging sites like ESRF, SLS or similar, where data are recorded on-site with remote access afterwards. This is probably mostly relevant for users with a certain experience in handling big data sets.

ERDA comes with low-level background import and export of data through different efficient transfer protocols like SFTP/FTP(S)/RSYNC/HTTP(S) and WebDAV(S).

From your Files page you can click the small data transfer icon ( ) top right in the address bar.
to open the data transfers page. Alternatively you can right-click a folder on the Files page and select Data Transfers and then pick Import or Export.

The default Manage Data Transfers tab on the data transfers page is used to set up and monitor transfers. On the Manage Transfer Keys tab it is possible to set up transfer keys, i.e. digital keys used for a more secure remote login than a simple login with password. Let’s see how one creates such an ssh-key and use it to import data from a server with an existing remote SFTP login.

**NOTE**: in case remote login with a password is allowed, one can simply skip these steps to create an ssh-key and just pick Login with password instead of Login with key when creating a transfer.

We go through the steps of creating a key from the Manage Transfer Keys tab. The name of the key is not that strict and it's only really used to point out the correct key when setting up the transfer afterwards.
After entering a name for the key and clicking the **Generate Key** button one is presented with the public part of the new key pair. This public key is shown along with some instructions on how to use it. In short one needs to copy the public key to the server where it is to be used for login with the private key.

**NOTE:** the private part of the ssh-key is **only** saved in an inaccessible location on ERDA, so nobody has access to it – and in turn to any machines where you allow it as a means of remote login.
Once you've inserted the public key on the import or export target host as instructed, you can proceed with setting up a background transfer to or from your ERDA folders. In this case it is depicted how a data import with SFTP from the host marge.esci.nbi.dk is set up with username bardino and with the recently created ssh-key, which we named id_rsa-marge. The transfer is configured to import the contents of the build/circular/ folder from the remote host and save them in the dataimports/ folder on ERDA.
A click on the Request transfer button enqueues the transfer
and you can monitor the progress in the overview and with the associated status buttons to the right there.
The transfer can be monitored more closely in the destination folder by clicking the local component-icon ( ), and in this case the result of the import looked like this, with a few files and folders recursively imported from marge.esci.nbi.dk.
All the other possible transfer combinations follow the same general procedure. If a transfer fails you can find some information about what might have gone wrong through the status files-icon ( atenção). Sometimes it is tricky to debug exactly what went wrong with the remote login or transfer, so feel free to contact us for help.
Courses and External Collaboration

ERDA is not only for internal data storage and sharing, but can also be an asset for courses and for collaboration across organizational boundaries - be it with domestic or foreign partners. In addition to data sharing, e.g. the interactive data analysis facilities like Jupyter can be a useful means of getting better acquainted with a particular subject, to let theory and practice go hand in hand. That is, it can be used both to present content and by giving participants an understanding of it, by letting them experiment and work with it interactively.

External Users / Peers

Research is rarely a purely local activity only happening at the university, and in a significant number of projects easy data sharing with other domestic or foreign participants from academia and private companies is therefore mandatory. ERDA is fundamentally designed with such considerations as a central requirement. Thus, access is not technically restricted by outdated castle-and-moat approaches, requiring users to either physically be “inside a perimeter” at UCPH or mimic it by means of VPN or similar. Instead ERDA relies on a zero trust security model throughout, with strictly secure access from anywhere.

Of course we cannot just let anyone get access to arbitrary data and storage. So data isolation is applied at a users and workgroup share level, and external users can only get an ERDA account if an employee at UCPH explicitly mandates their access. Management requires a formal confirmation to be in place for the latter. We have chosen to integrate that directly in ERDA with the Peers functionality, where employees can specify whom they need to collaborate with or provide access to for course/workshop purposes.

As an employee you have two overall choices available for that

1. You specify peer details with the option to send out a matching invitation
2. You let them sign up themselves and then accept their pending user account creation

Externals need to use our stand-alone login solution with their email address as username and a personal strong password of choice. Therefore as such they need to fill a short sign up form choosing their password. Using (1) the sign up form is pre-filled with all other fields you entered, whereas in (2) they have to fill the corresponding ID fields themselves and specifically reference you as their contact employed at UCPH – in order for you to accept their account. That is, they must write in the Comment field that it’s for a collaboration or course with you as contact, and preferably mention your UCPH email address there, too.

An example inviting a former colleague for a few years of on-going collaboration could look like

4 https://www.nist.gov/publications/zero-trust-architecture
Checking *Invite on email* before saving results in an automatic invitation email to the entered user(s), including a link to a personal pre-filled sign up form. In that way it’s only needed for them to choose a personal password and accept the user terms before clicking *Send*.

Note: *Label* is just a simple name of your choice to help you keep track of your peers. It does not allow space or various special characters, but you may use ‘-’ or ‘_’ to separate words.
The ERDA administrators will then create the user account and thereby automatically send a brief welcome email with login details, etc.

As an employee you can use Show Peers to view and manage your external contacts, plus send out invitations in case you skipped that during registration.
The ERDA administrators use the given Expire value to time-limit access to the external user accounts. In practice the access always lasts at most one year at a time to limit stale accounts, but with the option to renew access without repeated confirmation from the employee as long as it doesn’t exceed the Expire date. After the expire date such renewal requires updated or repeated peer acceptance.

In case you want to change an Expire value later on, you can do so by entering the same user ID field values under Enter Peers or Import Peers, pick Update in the Actions drop-down and choose another Expire value.

**Personal Copy of Material / Import Sharelink**

For collaboration purposes it usually makes most sense to use Workgroups with the associated shared folders for data sharing. In that situation there’s only one shared version of the data for all to work on.

In teaching contexts, on the other hand, it is common to want personal copies of course/workshop material for participants to work on. E.g. if the material involves Jupyter notebooks or similar that they are supposed to individually extend as exercises. A common approach in such situations is to make a folder with the relevant material and create a read-only share link on it. Then distribute the share link or just the ID of it to the participants, instructing them to create a folder in their ERDA Files and use Share Link → Import from the right-click menu to fetch their own personal copy to work on.
In the resulting import dialog they enter either the handed out Share Link ID or the complete URL of it. In the example here it’s shown for a link with ID *ddYihYUKMP*. 
When clicking **Ok** the contents of the share link are copied to the chosen folder, where it’s ready for use in exercises and so on.
Support, Questions and Comments

We welcome all comments on how we can improve the user experience as well as any questions you may have about the use of ERDA. This includes support in case something doesn't work the way it is intended and described here.

It is very helpful for us if you include information about your operating system and browser with any support requests. If you aren't sure you can open up

https://www.whatismybrowser.com/ or https://detectmybrowser.com/

and send us at least the line below “Your web browser is:” in the blue block at the top or the line starting with “You're using” respectively.

Screenshots of any problems are also very useful.

Feel free to contact us by sending an email to ERDA Support <support@erda.dk>.

Kind Regards, the UCPH ERDA team.