

Getting setup on Flywheel (Mac)

v1.1 ~ Jan 2024

Web login

A Google account needs to be linked in order to access the Flywheel platform. (Check with site admin). The Gmail needs to be all lower case. Once this is done you can access the site by going to our unique URL below and entering your Google credentials.

(The following links may be useful to save in the bookmark bar of a web browser)

Login link (<https://bmgf.flywheel.io/>)

Flywheel documentation <https://docs.flywheel.io/hc/en-us>

When logged into our dedicated site. There will be a list of projects you may have access to. If there is a project that you cannot see that you believe should be there contact a system admin to grant permissions to that project.

Uploading data

Setup

There are two ways to upload data.

1. via the web console
2. the command line interface (CLI)

Files can be dragged and dropped into the web console. This is useful for small amounts of data.

The command line interface is the preferred option, particularly when handling large amounts of data. To do this some dependencies need to be installed on your local machine so it can talk to the Flywheel platform.

There are detailed instructions provided here:

<https://docs.flywheel.io/hc/en-us/articles/360008162214-Installing-the-Flywheel-Command-Line-Interface-CLI->

An IDE (integrated development environment) is recommended to run the CLI. This is a program that allows you to run code and scripts. The IDE we recommend is VS Code. This can be downloaded here:

<https://code.visualstudio.com/download>

Step 1: Download

In the upper-right corner, select your account menu, and select Profile. Here there will be the option to download the CLI for your system.

| Projects | | Subjects | Sessions | Role | Users |
|------------------------------|------------|----------|----------|-------------------|-----------------------|
| Project | Group | | | | |
| AKU (Neurodevelopment) | Global Map | 0 | 0 | read-write | SD PV RM |
| Bangladesh (REVAMP) | Global Map | 0 | 0 | read-write | PV |
| beta_UCT_D2 | dev | 17 | 50 | admin | |
| beta_UCT_Khula | dev | 23 | 23 | admin | |
| Brown (RESONANCE_highField) | Global Map | 0 | 0 | read-write | SD NL PV SK RM |
| Brown (RESONANCE_lowField) | Global Map | 115 | 132 | read-write | DD SD NL RB SK +1 |
| Ethiopia (ENAT) | Global Map | 0 | 0 | read-write | SD PV |
| Ghana (Accra) | Global Map | 0 | 0 | read-write | SD FO PV |
| India (CEL_Neurodevelopment) | Global Map | 0 | 0 | read-write | SD NL PV |
| India (RAPIDIron) | Global Map | 0 | 0 | read-write | DD SD PV |
| KCL (DenmarkHill) | Global Map | 0 | 0 | admin, read-write | SD PV DD SK RM |
| KCL (StThomasHospital) | Global Map | 0 | 0 | admin, read-write | Pablo Velasco (admin) |
| Malawi (REVAMP) | Global Map | 0 | 0 | read-write | RB SD PV |
| Methods development | Global Map | 0 | 0 | admin, read-write | SD PV |
| test_upload | dev | 4 | 4 | admin | |

<https://bmfg.flywheel.io/#/projects/62debe047e3d6be32667cfab/permissions>

1 – 22 of 22

If on a Mac Flywheel will need to be added as a security exception in system preferences before you can open. *If on a mac pro (M1 max chip) it may be necessary to start in safemode and allow user management of system terminal extensions under security options.*

Step 2: Generate an API key

Also under Flywheel Profile tab (Do not share)

Step 3: Navigate to the fw.exe or fw app

Move to somewhere it can be run from. Ideally on your system path. See Flywheel documentation in link above for more information.

From terminal on Mac (open by pressing command + space and start typing Terminal)

```
sudo cp ~/Downloads/darwin_amd64/fw /usr/local/bin/
```

Step 4: Log in from the CLI

Should be possible to login via the command line with the site address and your API

```
fw login [url].flywheel.io:[PERSONAL_API_KEY(DO NOT SHARE)]
```

Sending local data to Flywheel via the CLI

This is an example of the folder hierarchy of data as it comes from the Hyperfine scanner (note: acquisition names will vary). The user provides the subject ID at the scanner. This should not contain any special characters and should contain leading 0s.

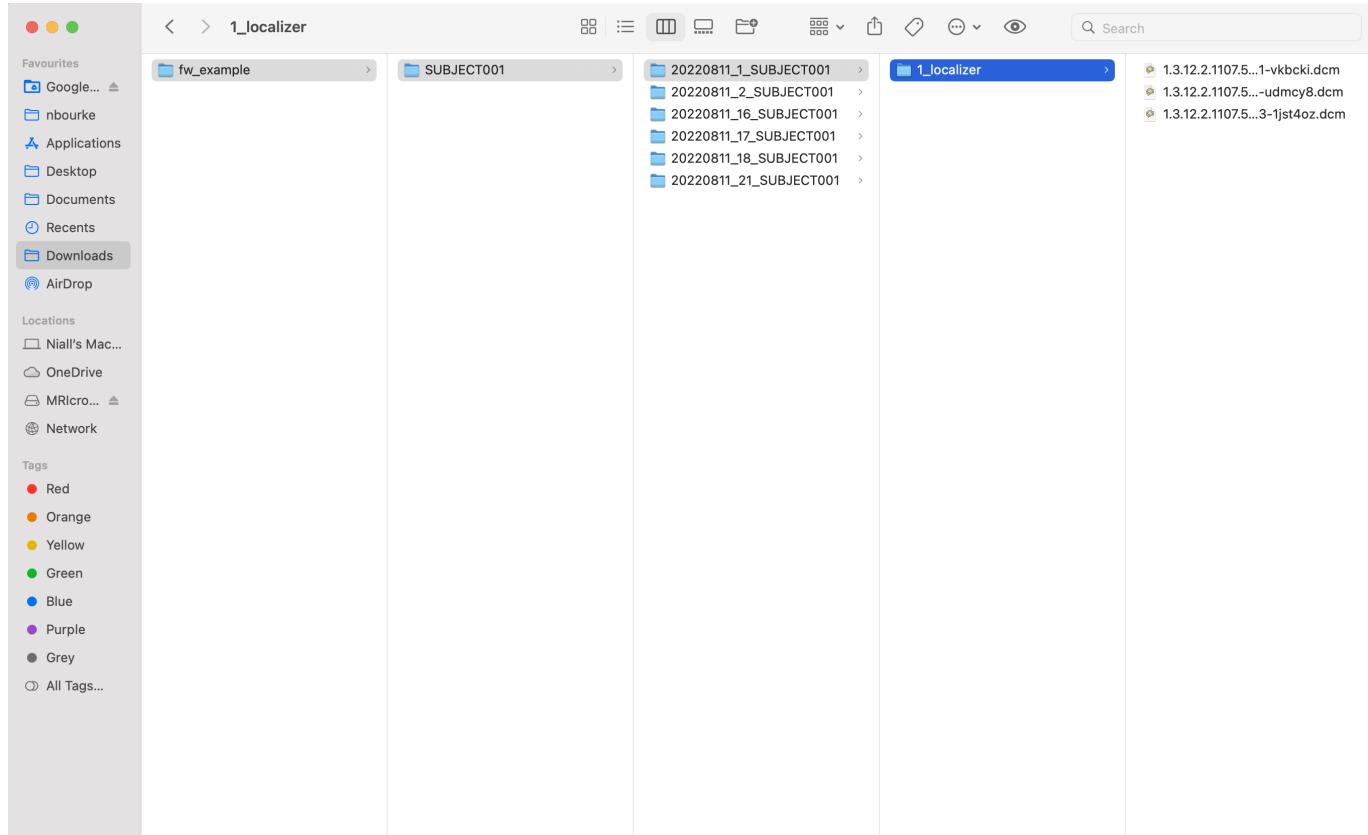
for example:

- BETA001
- GAMMA001

- ECHO001

not:

- BETA_1
- GAMMA-001
- ECHO(01)



On the command line

First you need to open terminal and navigate to where the data is stored, then;

Connect/login to the CLI (command line interface) and finally;

Run the import command pointing to the folder where the data is

Open terminal by pressing command + spacebar

Start typing: Terminal

```
Last login: Thu Aug 11 09:19:11 on ttys002
(base) nbourke@Nialls-MacBook-Pro Downloads % pwd
/Users/nbourke/Downloads
(base) nbourke@Nialls-MacBook-Pro Downloads % ls
fw_example
(base) nbourke@Nialls-MacBook-Pro Downloads % fw login bmgf.flywheel.io [REDACTED]
You are now logged in as: Niall Bourke!
(base) nbourke@Nialls-MacBook-Pro Downloads % fw ingest dicom --detect-duplicates fw_example dev "example"
Created [2022-08-11 10:40:16]

Configuring [2022-08-11 10:40:16]
100.0% (1s)

Scanning [2022-08-11 10:40:17]
510/510 files, 422MB (1s)

Resolving [2022-08-11 10:40:18]
100.0% (1s)

Detecting duplicates [2022-08-11 10:40:19]
100.0% (0s)

In review [2022-08-11 10:40:19]
Hierarchy:
Groups: 1
Projects: 1
Subjects: 1
Sessions: 1
Acquisitions: 6
Files: 0
Packfiles: 6

Confirm upload? (yes/no): yes

Preparing [2022-08-11 10:40:24]
100.0% (1s)

Uploading [2022-08-11 10:40:25]
100.0% - (0 failed) (46s)
Total: 6

Finalizing [2022-08-11 10:41:11]
100.0% (1s)

Finished [2022-08-11 10:41:12]

Final report
Total elapsed time: 56s
(base) nbourke@Nialls-MacBook-Pro Downloads %
```

pwd = print working directory. Shows where on the computer path you are located

ls = list contents of current directory. Here we are in Downloads and there is one folder "fw_example"

fw login bmgf.flywheel.io.xxxxxxx = Flywheel login command we have set up. This contains the users unique key at the end which should not be shared. When this is run successfully it will return a line saying you are logged in.

fw ingest dicom --detect-duplicates fw_example dev "example" = This line sends the data inside the folder fw_example to the project example in the group dev on the Flywheel platform. For sending real data dev needs to be replaced by *global_map* and "example" needs to be replaced by *your project name*

After this command is run some output will be generated in the terminal. A review of the expected folder structure will be provided with the number of subjects and sessions. Have a look at this to check if it is what is expected. If so respond by typing "yes" to the prompt. The data will start uploading to the Flywheel platform.

The data will be uploaded to the Flywheel platform under the project and group specified(here group = dev, project = example). Opening up the Flywheel web browser we can see the subject and scans that have been uploaded. The subject label and time stamp are given on the left hand side and the list of scans are given on the right. In this example the dicom files from the scanner have automatically been converted to NIFTI the analysing file format. Going forward additional functionality will be added to curate and analyse the data on this platform as new data is added.

The screenshot shows the Flywheel web interface. On the left, there's a sidebar with navigation links for Flywheel, DATA, COMPUTE, REPORTING, and ADMIN. The main area is titled "example / dev (fw://dev/example)". It has tabs for Description, Sessions, Information, Analyses, and Data Views. The Sessions tab is selected. Below it, there are filters for Actions and Advanced Filters, and buttons for Download Project and Delete Project. The main content area displays a table of sessions. One session is expanded to show its details:

| | File Description | Created Date | Classification | All Types |
|--------------------------|---|------------------|-------------------|-------------|
| <input type="checkbox"/> | 1 - localizer 15:49 | 2022-08-11 10:40 | MR: Localizer, T2 | dicom |
| <input type="checkbox"/> | 1 - localizer_i00001.json | 2022-08-11 10:40 | MR: Localizer, T2 | source code |
| <input type="checkbox"/> | 1_localizer_i00001.nii.gz | 2022-08-11 10:40 | MR: Localizer, T2 | nifti |
| <input type="checkbox"/> | 1_localizer_i00002.json | 2022-08-11 10:40 | MR: Localizer, T2 | source code |
| <input type="checkbox"/> | 1_localizer_i00002.nii.gz | 2022-08-11 10:40 | MR: Localizer, T2 | nifti |
| <input type="checkbox"/> | 1_localizer_i00003.json | 2022-08-11 10:40 | MR: Localizer, T2 | source code |
| <input type="checkbox"/> | 1_localizer_i00003.nii.gz | 2022-08-11 10:40 | MR: Localizer, T2 | nifti |
| <input type="checkbox"/> | 16 - NODDI_b0 17:05 | | | |
| <input type="checkbox"/> | 16 - NODDI_b0.dicom.zip | 2022-08-11 10:40 | MR: B0, NODDI | dicom |
| <input type="checkbox"/> | 16_NODDI_b0.json | 2022-08-11 10:40 | MR: B0, NODDI | source code |
| <input type="checkbox"/> | 16_NODDI_b0.nii.gz | 2022-08-11 10:40 | MR: B0, NODDI | nifti |
| <input type="checkbox"/> | 17 - NODDI_b0_reversed 17:06 | | | |
| <input type="checkbox"/> | 17 - NODDI_b0_reversed.dicom.zip | 2022-08-11 10:40 | MR: B0, NODDI | dicom |
| <input type="checkbox"/> | 17_NODDI_b0_reversed.json | 2022-08-11 10:40 | MR: B0, NODDI | source code |
| <input type="checkbox"/> | 17_NODDI_b0_reversed.nii.gz | 2022-08-11 10:40 | MR: B0, NODDI | nifti |
| <input type="checkbox"/> | 18 - NODDI_90dir_9b0_multishell 17:09 | | | |
| <input type="checkbox"/> | 18 - NODDI_90dir_9b0_multishell.dicom.zip | 2022-08-11 10:41 | MR: B0, NODDI | dicom |
| <input type="checkbox"/> | 18_NODDI_90dir_9b0_multishell.bval | 2022-08-11 10:41 | MR: B0, NODDI | bval |
| <input type="checkbox"/> | 18_NODDI_90dir_9b0_multishell.bvec | 2022-08-11 10:41 | MR: B0, NODDI | bvec |
| <input type="checkbox"/> | 18_NODDI_90dir_9b0_multishell.json | 2022-08-11 10:41 | MR: B0, NODDI | source code |
| <input type="checkbox"/> | 18_NODDI_90dir_9b0_multishell.nii.gz | 2022-08-11 10:41 | MR: B0, NODDI | nifti |

Flywheel hosts webinars that provide a good introduction to the platform, including uploading data.

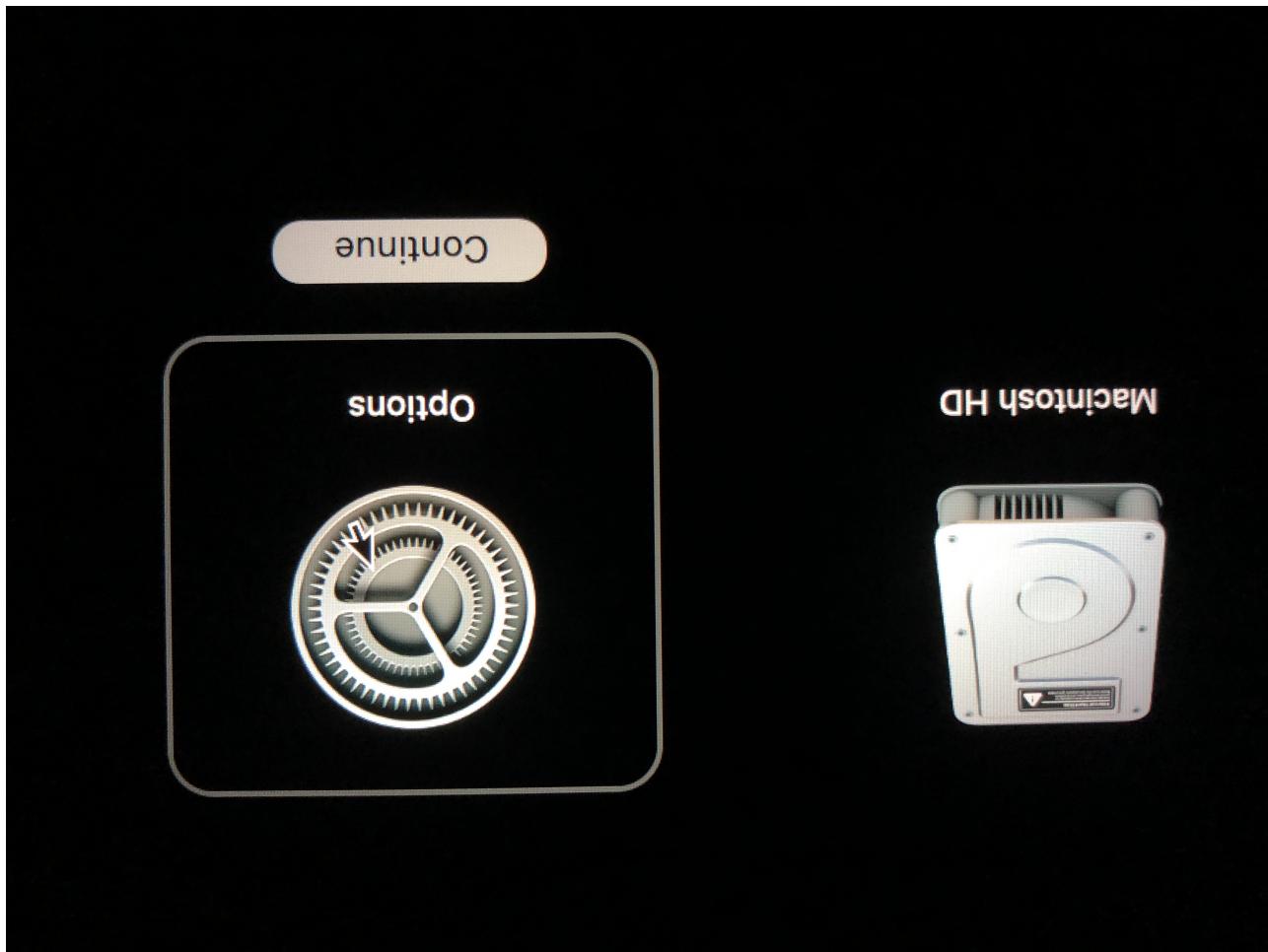
Mac troubleshooting Flywheel setup

Mac silicon chip security settings

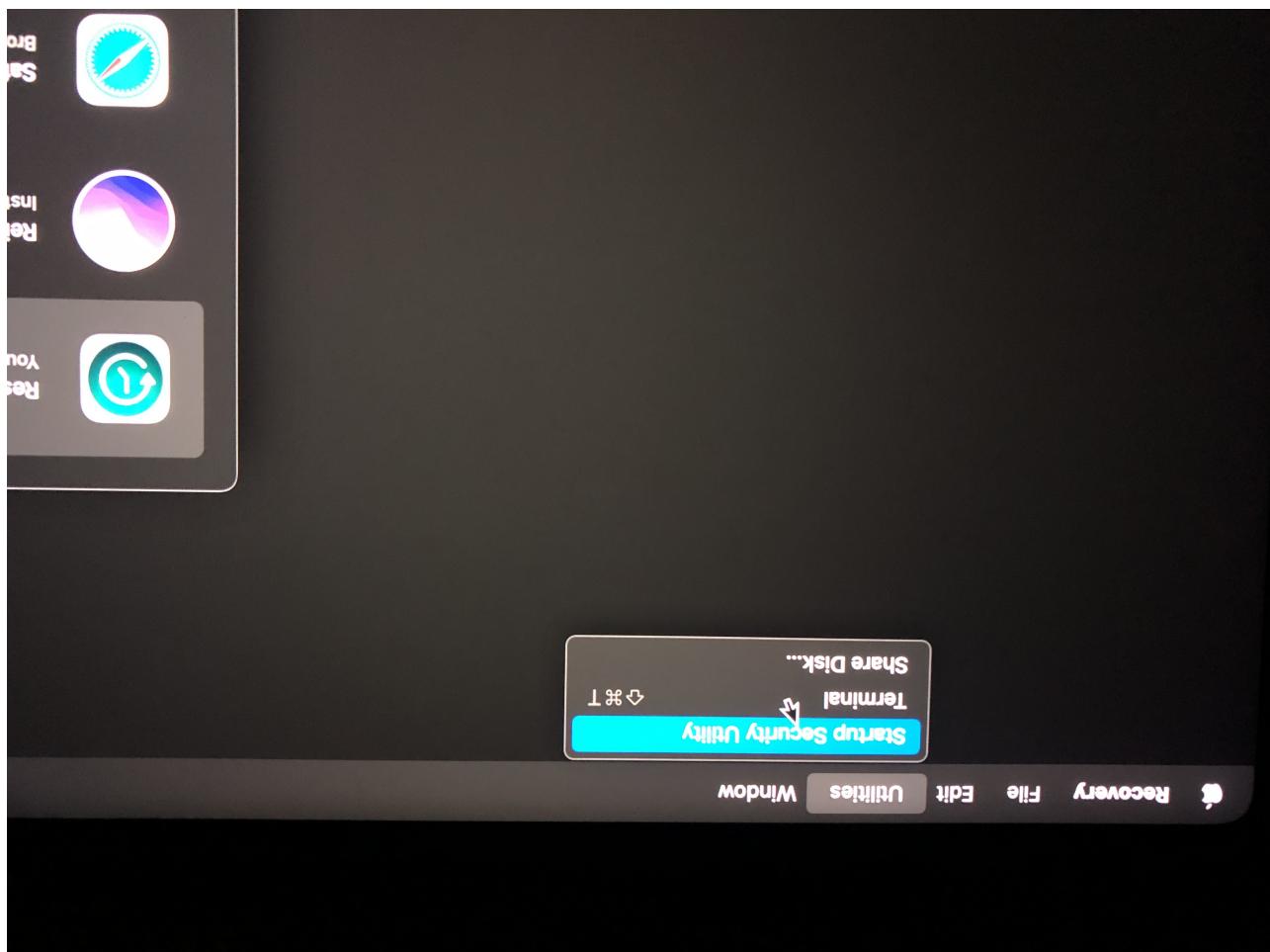
On the latest Macs with silicon chips it may be necessary to change the startup disk security settings to allow use of kernal extensions. To do so follow the instructions below. During the process an administrators details may be required.

1. Shut down the computer
2. Press and hold the power button. Keep holding and a message should appear "Loading startup options"

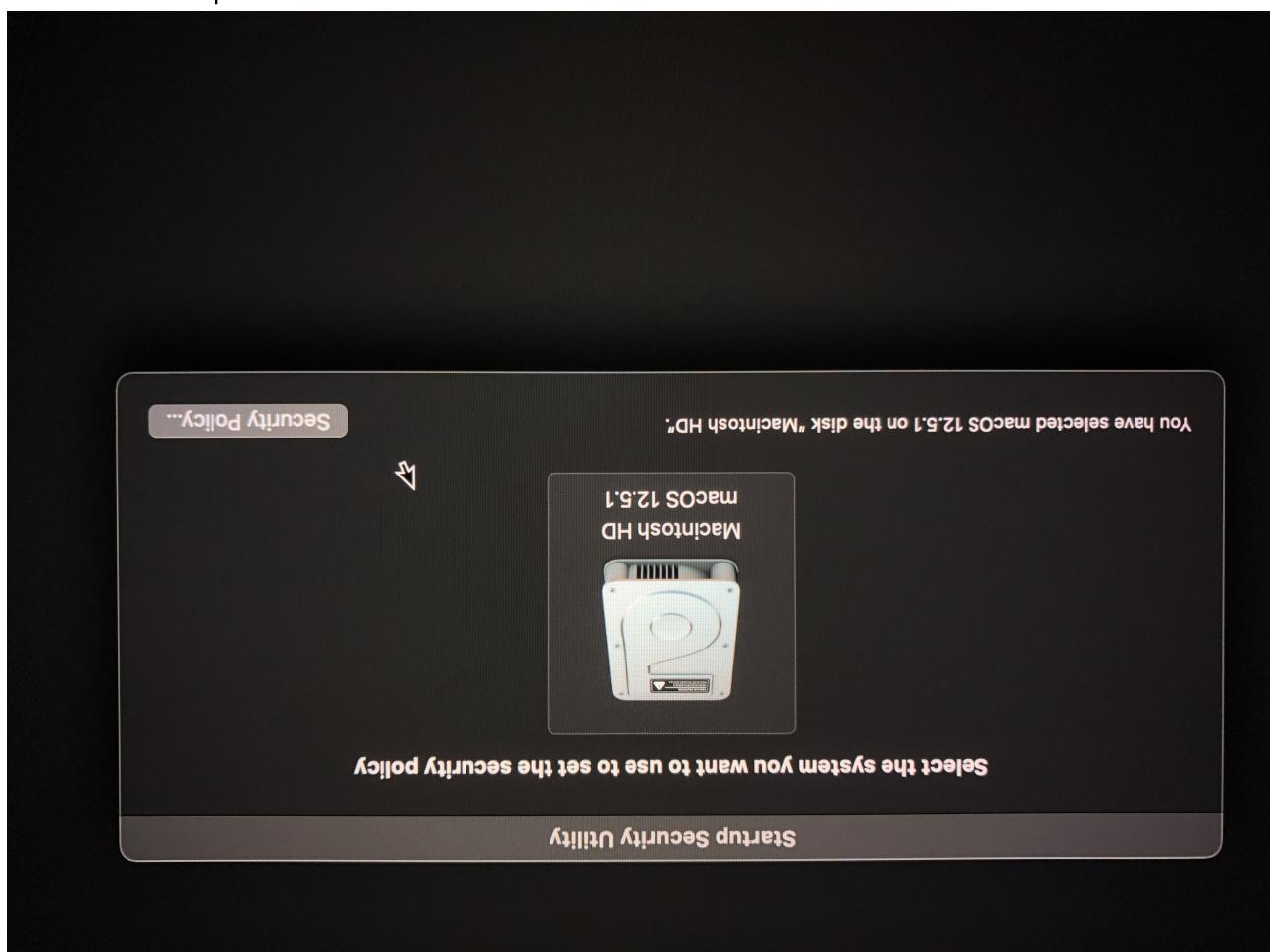
3. Select "Options"



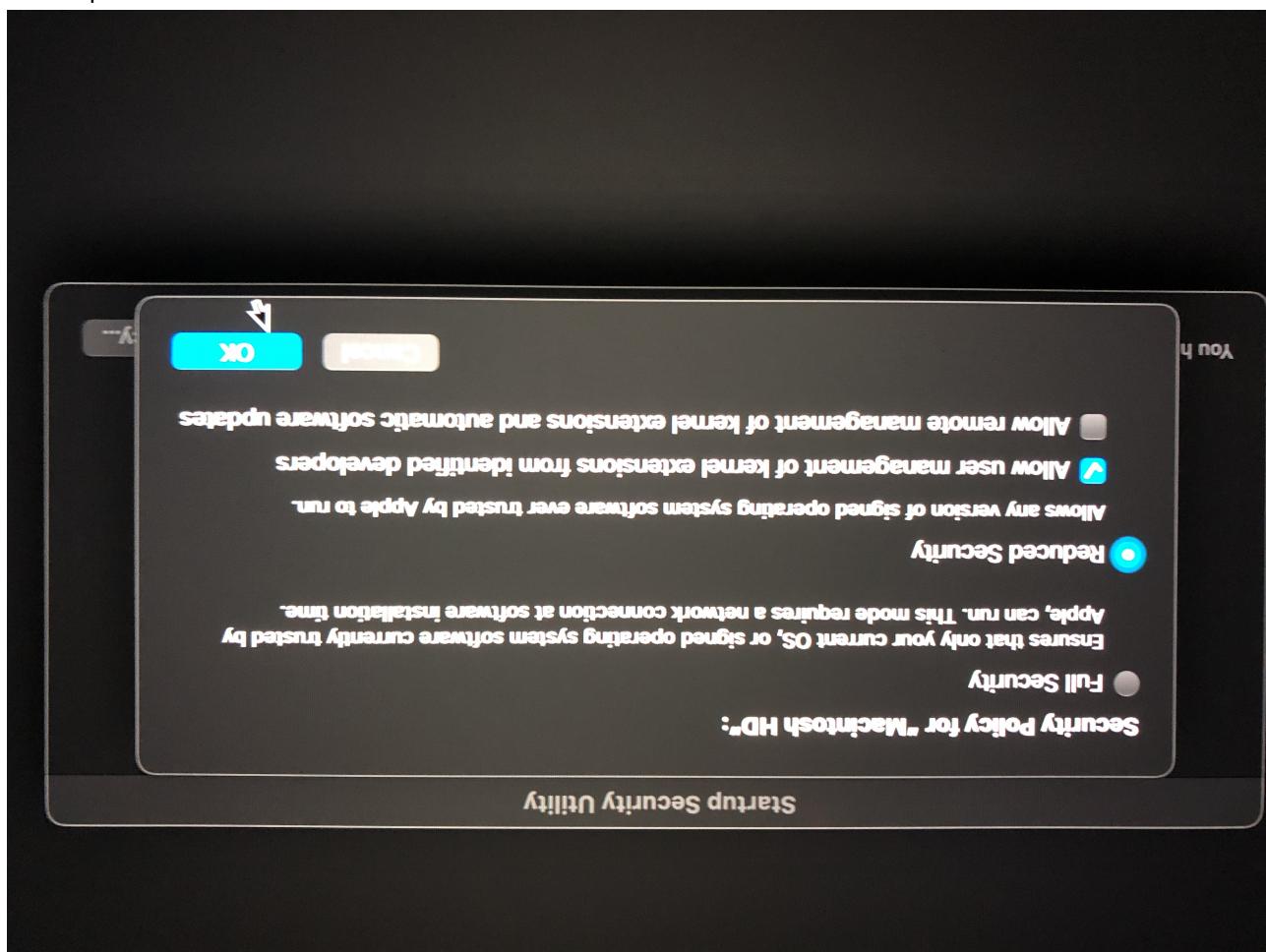
4. When the menu appears, ignore this. Select the Utilities > Startup Security Utility from the toolbar menu on the top left of the screen



5. Select the startup disk



6. Select "Reduced Security" & "Allow user management of kernel extensions from identified developers"



7. Click "OK"

8. When the changes are applied restart the computer and continue with installation.

In some instances it may be necessary to install Rosetta on Macs with the new silicon chip if it is not already available. This simply improves compatibility with software designed with intel chips.

1. Open Terminal (Command + space), type Terminal
2. copy and paste the following:

```
softwareupdate --install-rosetta
```

For more information: <https://support.apple.com/en-us/HT211861>