

Hydro-JULES Summer School DataLabs sessions

13th -15th July 2021

Welcome to the Hydro-JULES Summer School DataLabs sessions.

On Tuesday 13th July at 11:00 we will be describing the Hydro-JULES DataLab with live presentations and practical help. Please wait in the DataLabs room at this time to follow the session.

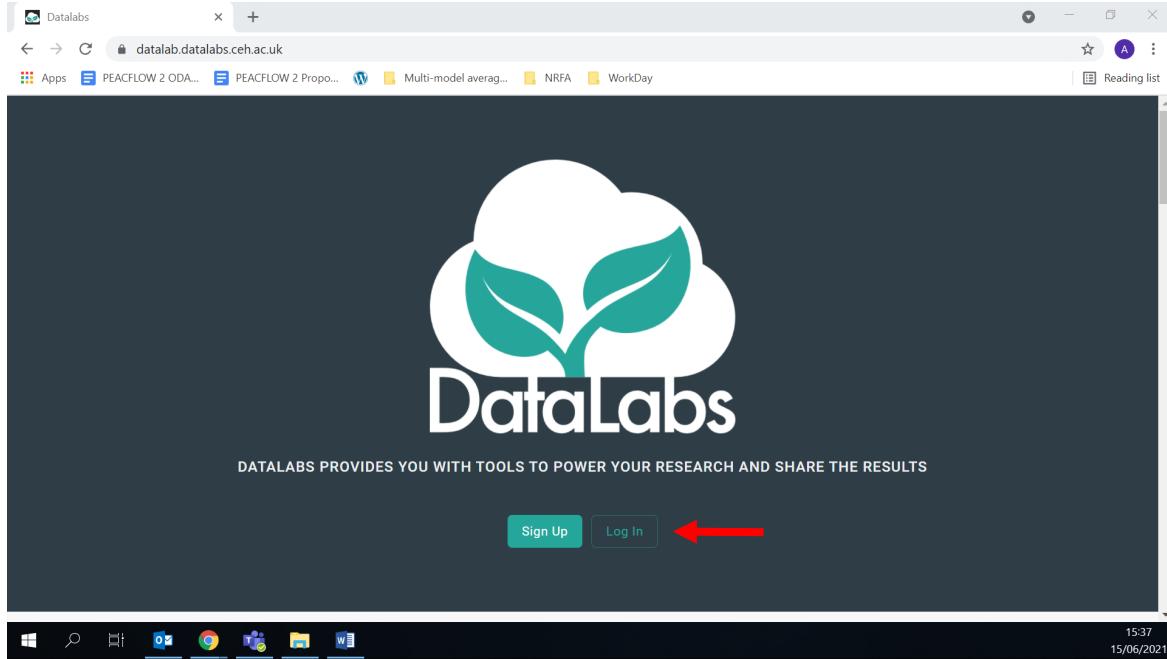
If you miss this session, there are a number of video presentations placed around the room which you can follow instead.. We will be running through these live during the morning sessions on Tuesday, Wednesday and Thursday, but you can view them at any time.

They cover the following topics (direct links to videos also provided):

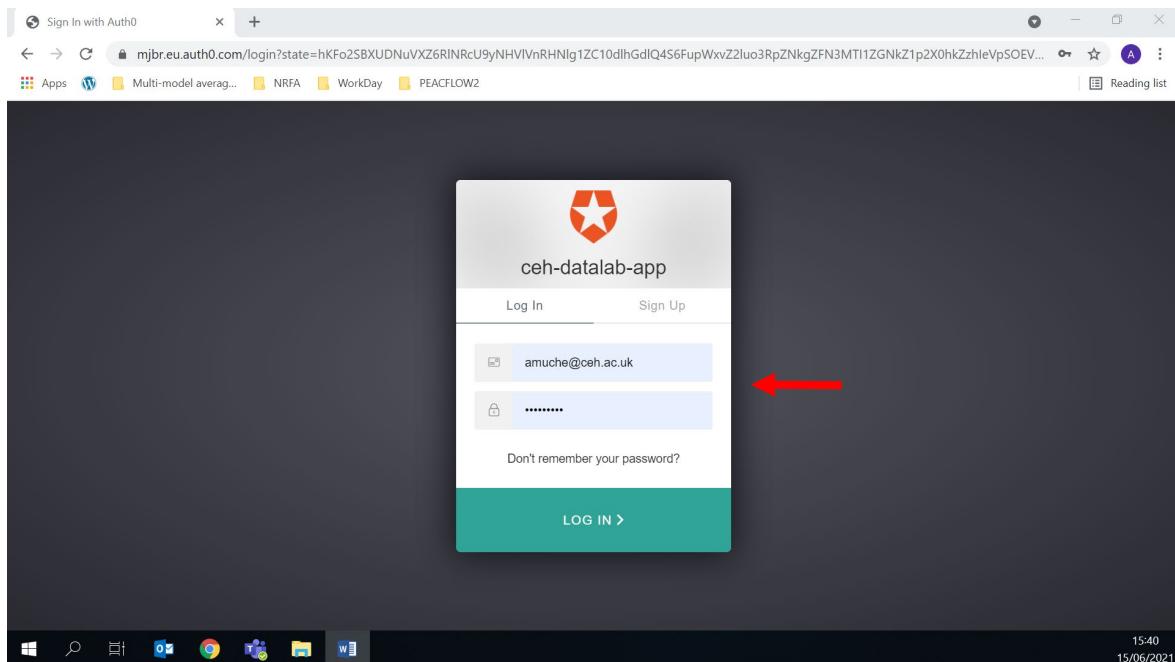
1. DataLabs overview: <https://www.youtube.com/watch?v=n68X8J4gi6Q>
2. netCDF access tutorial: <https://youtu.be/PktTVnQcQy4>
3. Cf-python examples: <https://youtu.be/dGif03kApJE>
4. Plotting observed data against modelled <https://youtu.be/kqez7RtCKdk>

This document provides a step-by-step guide to logging into and using DataLabs, and should be used alongside the notebooks and videos.

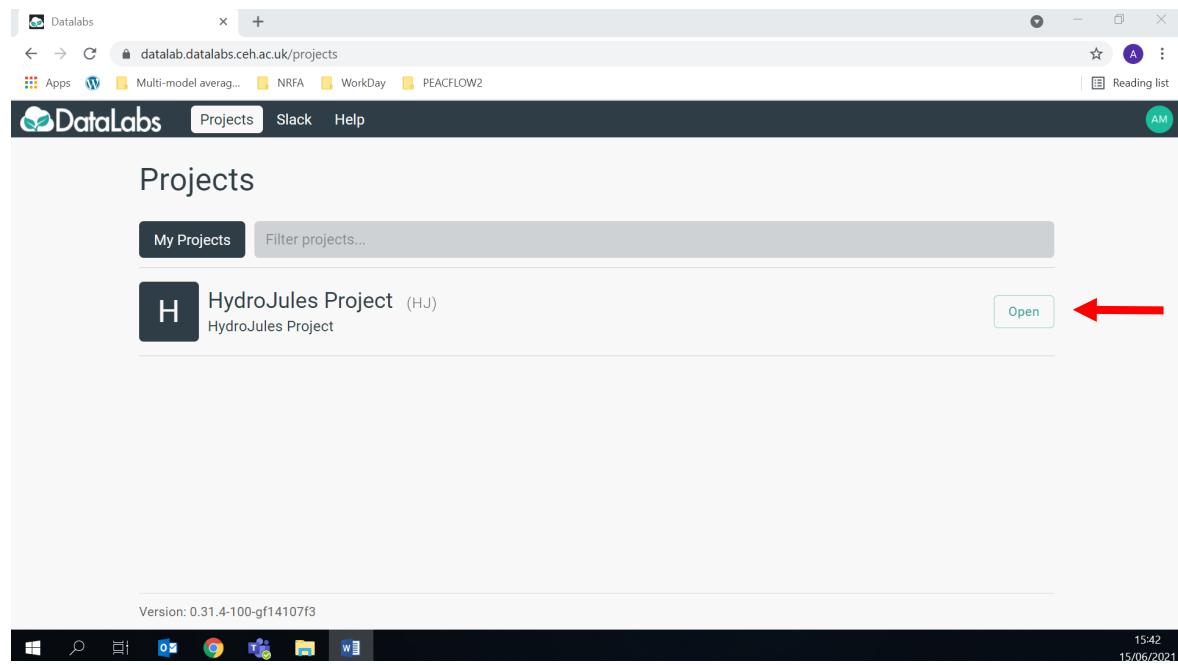
1. Please go to the webpage <https://datalab.datalabs.ceh.ac.uk/>, and click on the “Log In” button shown with the red arrow.



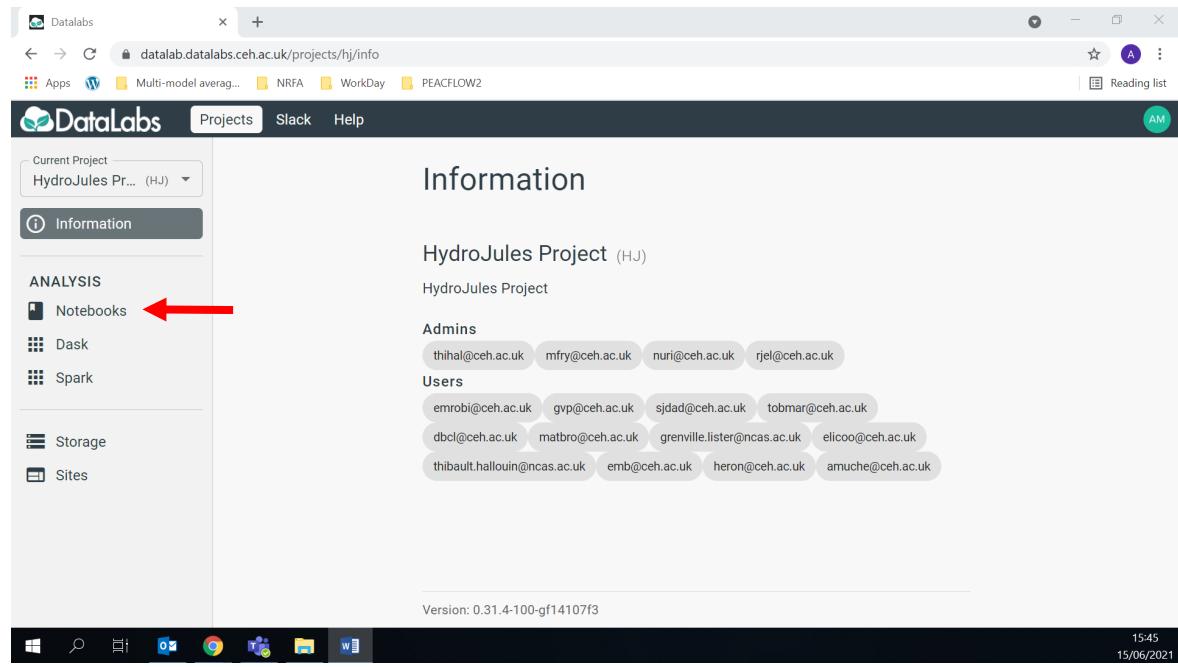
2. Please log in to the ceh-datalab-app (shown with the red arrow) with your created username and password.



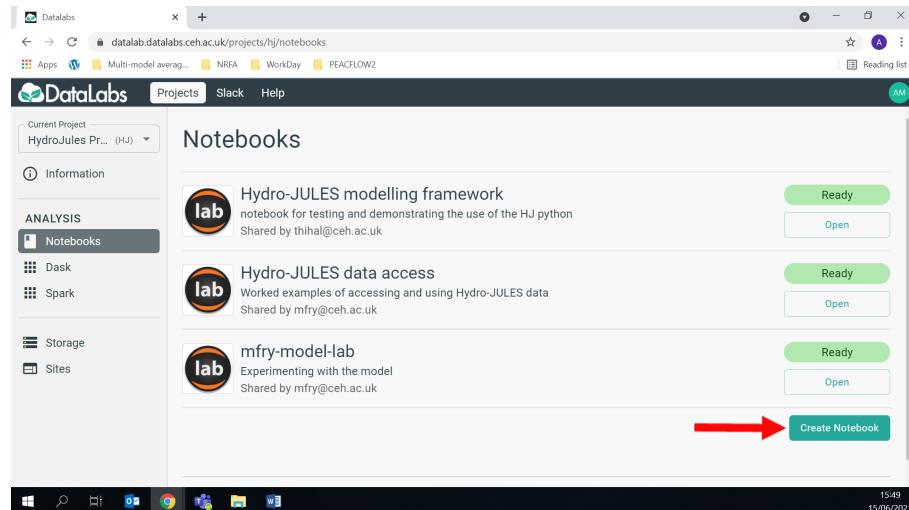
3. Click on “Open” tab for the HydroJules Project shown with the red arrow.



4. Please click on “Notebooks” tab (shown with the red arrow), in the left hand side panel under “Analysis” section.

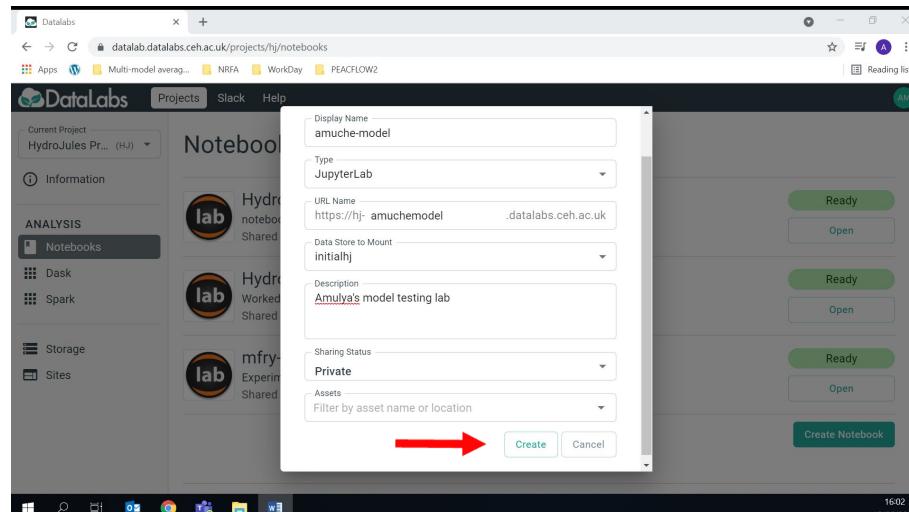


5. Please click on the “Create Notebooks” tab, on the right side of the webpage, shown with the red arrow. This will actually create a JupyterLab for you, which can host multiple notebooks. Fill in the options as per step 6 below

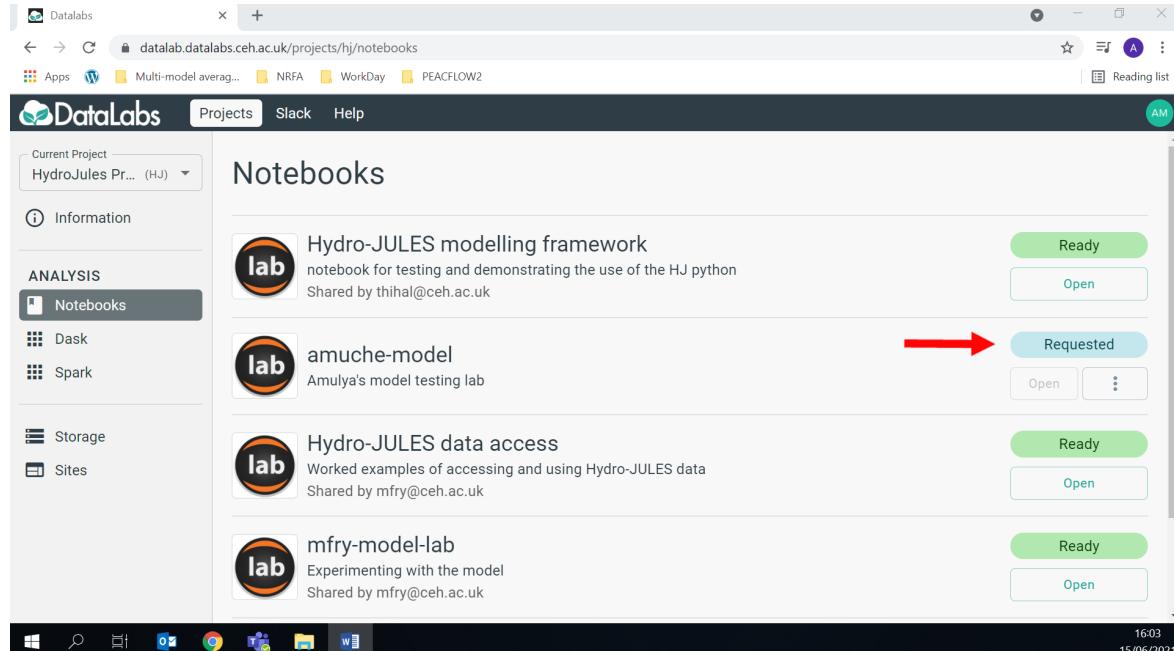


6. Fill in the following information in the pop up, scroll down if required.
- Display Name – *anything* that you like
 - Type – select “JupyterLab” option
 - URL Name – *any word* (ideally similar to the Display Name without any space, dashes or underscore)
 - Data Store to Mount – select “initialhj” option
 - Description – *a few words* describing the JupyterLab
 - Sharing Status – select “Private” option
 - Assets – please leave *empty* or do not select any option

Then click on the “Create” tab at the bottom, shown with the red arrow.

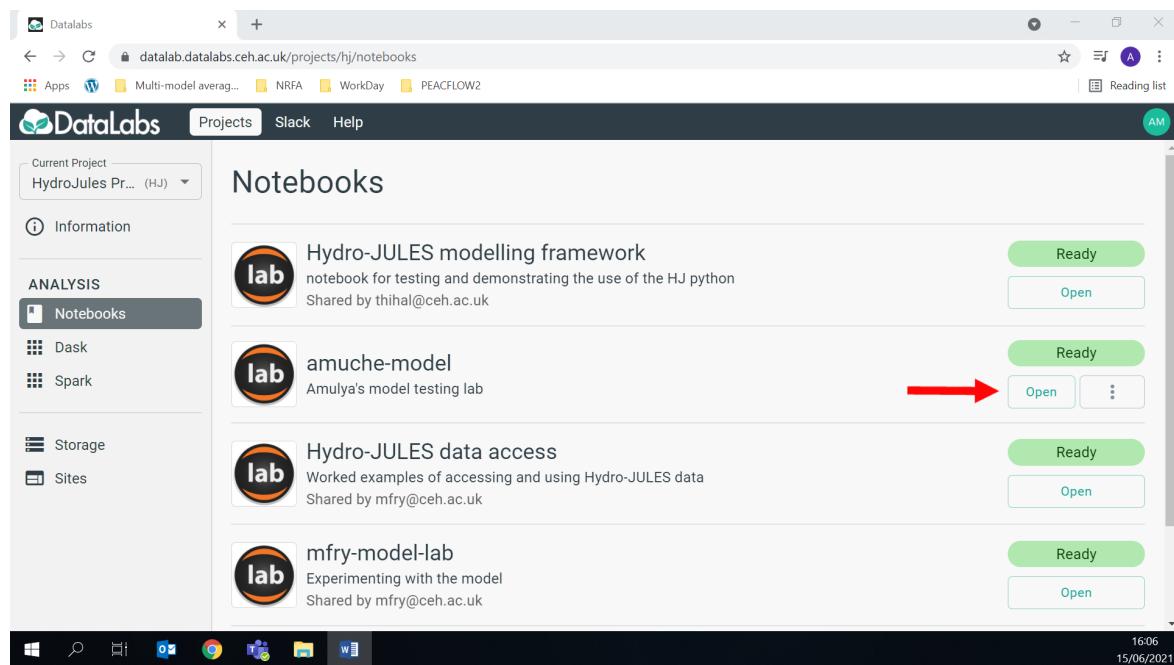


7. You should then see a new lab in your “Notebooks” page, with a blue “Requested” tab, shown with the red arrow. Please wait, this should turn into a green “Ready” tab in a few minutes (you may need to refresh your browser window).



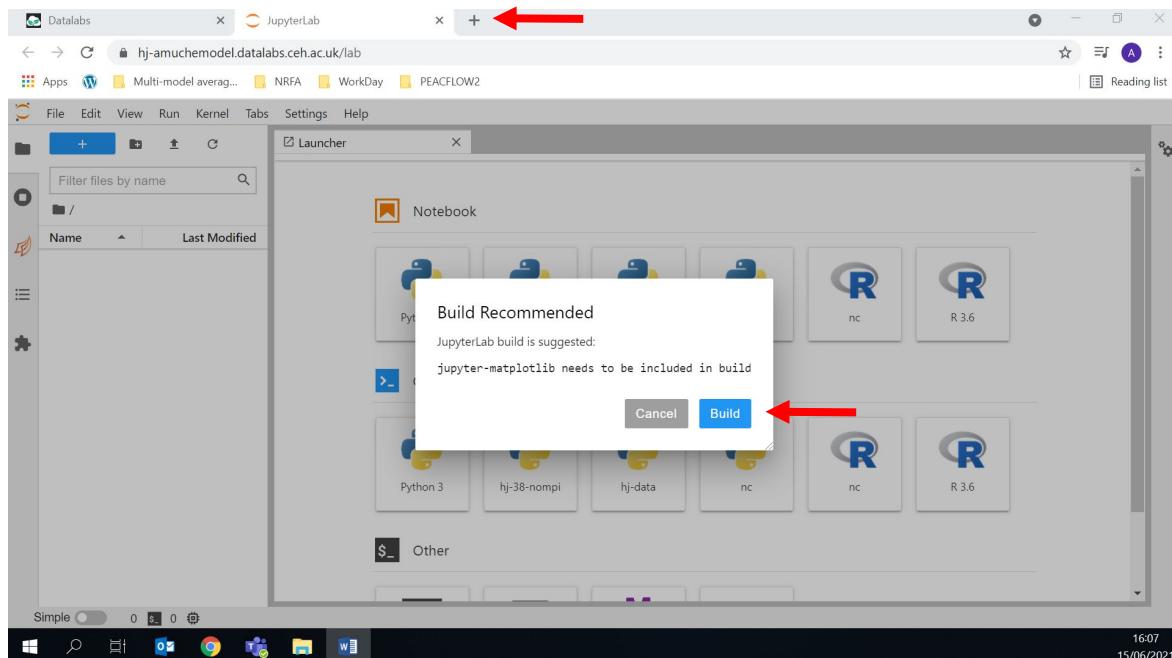
The screenshot shows the DataLabs interface with the 'Notebooks' tab selected in the sidebar. A new entry titled 'amuche-model' is listed, showing its details: 'Amulya's model testing lab'. To the right of the entry, there is a 'Requested' button, which is highlighted with a red arrow. Other entries include 'Hydro-JULES modelling framework', 'Hydro-JULES data access', and 'mfry-model-lab', each with a 'Ready' button.

8. Please click on the “Open” tab (shown with the red arrow), associated with your newly created lab.

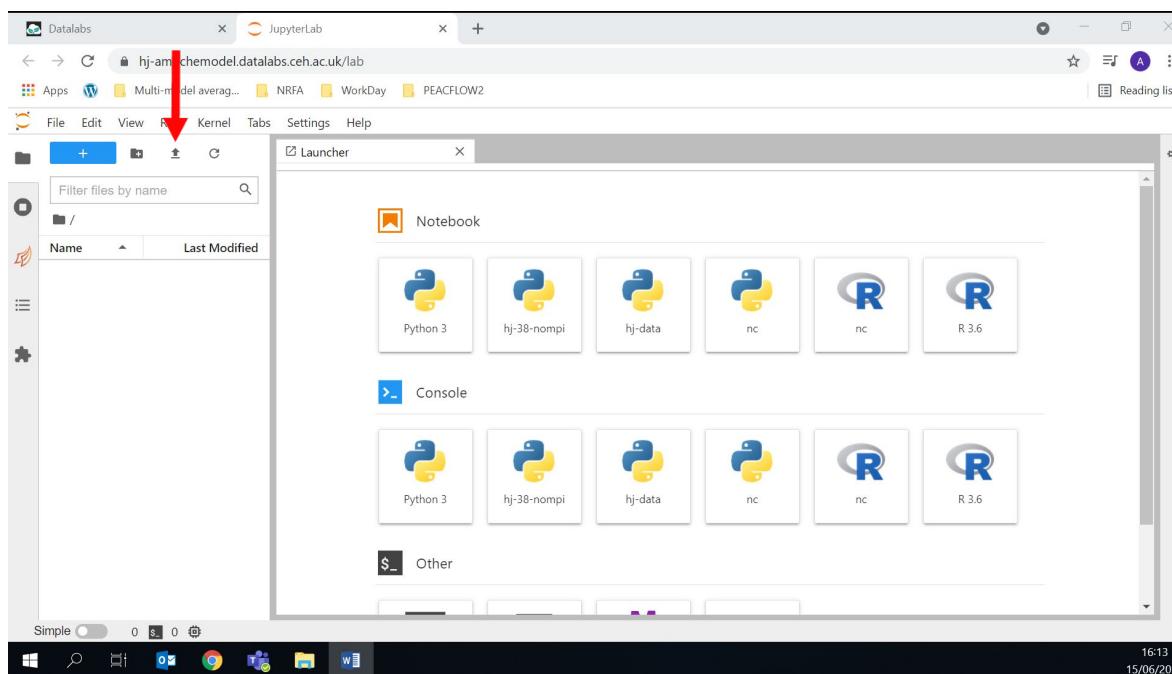


The screenshot shows the same DataLabs 'Notebooks' page after the 'Open' button for the 'amuche-model' lab has been clicked. The 'Requested' button has now turned green and is labeled 'Ready'. The red arrow from the previous step points to this 'Open' button.

9. This opens a new browser tab with your JupyterLab (shown with a red arrow). You will see a “Build Recommended” pop up. Please click on the “Build” button shown with another red arrow.

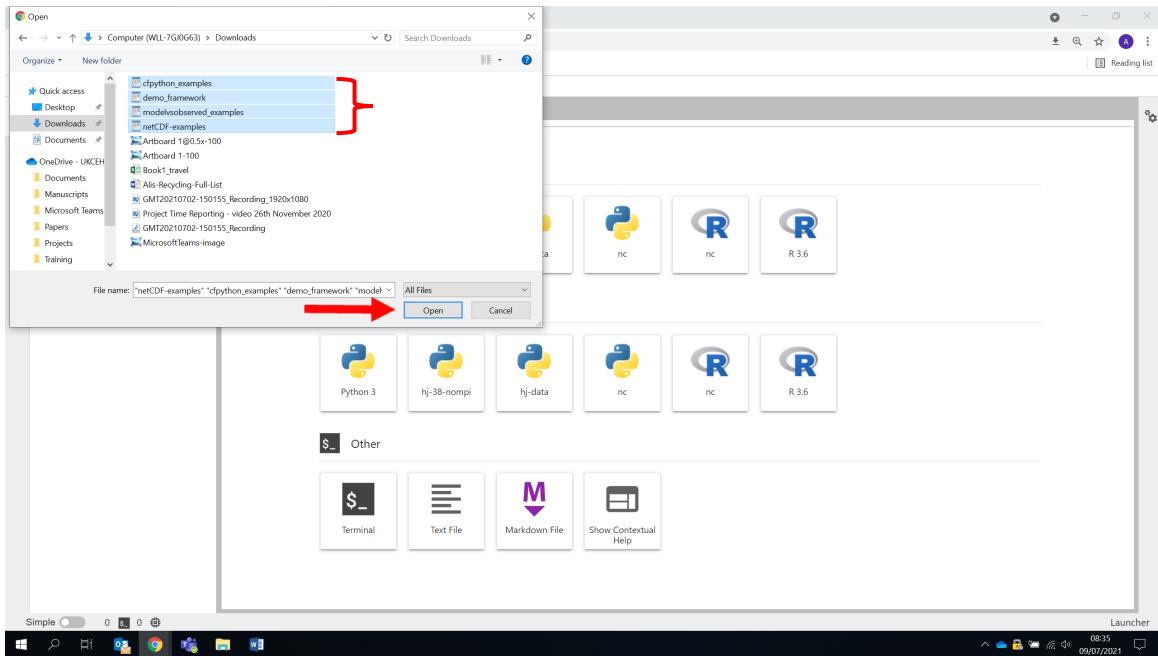


10. Please click on the upload icon shown with the red arrow.



11. Using the upload pop up, please go to the folder where the Jupyter notebooks that were sent to you are stored on your computer. Please select

all the 7 files (shown with the red bracket) and click on the “Open” tab shown with the red arrow.



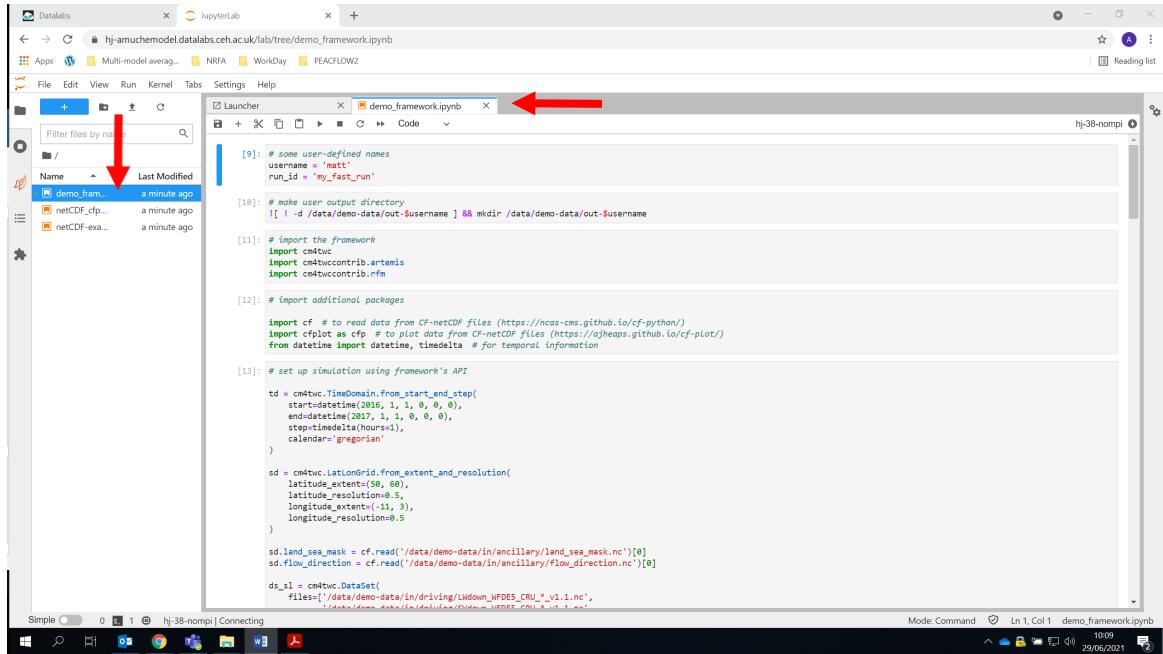
12. Now you can see all the 7 files in your lab's left hand panel. You can double click on any notebook to launch it. The suggested order for use of these notebooks is:

- 1) netCDF-examples.ipynb - this notebook walks through the structure of a netCDF format data file, and shows how to access data, and plot it, using the netCDF4 library
- 2) cfpthon_examples.ipynb - this notebook demonstrates how to use the CF-Python library, which has a number of useful additional features for use with “CF compliant” netCDF files.
- 3) cm4twc_demo.ipynb - this notebook walks through running a simple set of hydrological models using the Hydro-JULES modelling framework
- 4) cm4twc_exercise1.ipynb – this contains an exercise on using the modelling framework
- 5) cm4twc_exercise2.ipynb – another exercise on the modelling framework
- 6) modelvsobserved_examples.ipynb – this notebook demonstrates how to compare observed station data (in csv format) against modelled gridded dataset (in netCDF format)
- 7) exercise_answers – this notebook contains some answers to questions set in cfpthon_examples and modelvsobserved_examples notebooks

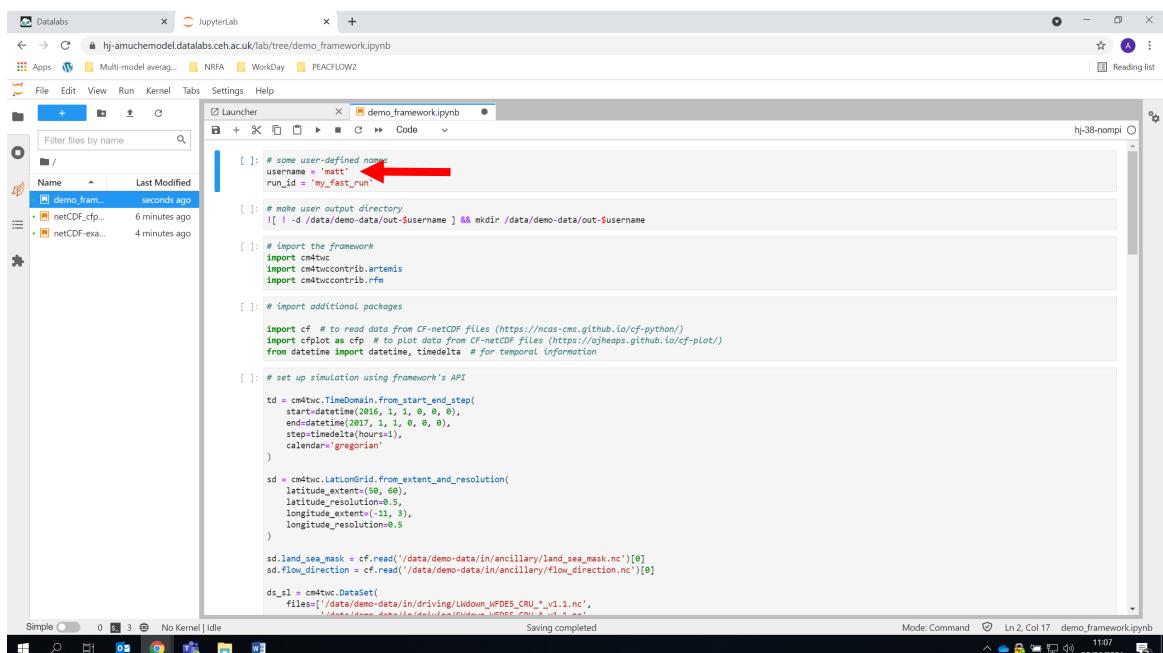
Please see step 16 for further useful hints about running Jupyter Notebooks.

13. When using the `demo_framework.ipynb` notebook there are a number of specific steps to undertake:

- Please double click on the “`demo_framework.ipynb`” notebook to launch it, shown with red arrows.



14. In the launched `demo_framework` Notebook, please go to second line of the Notebook and change the username from ‘matt’ to your “userid”. For example, with a username like “testing@ceh.ac.uk” the userid becomes “testing” (i.e. avoid the use of the @ symbol). Please input your own userid not “testing”.



15. After changing the username (shown with a red arrow), please save the Notebook by clicking on the save icon shown with another red arrow. To run the code please use ► (red box) which restarts the kernel and runs the whole code. Please see step 16 for details about running Jupyter Notebooks.

```
# some user-defined names
username = 'amuche'
run_id = 'my_fast_run'

# make user output directory
! [ -d /data/demo-data/out-$username ] && mkdir /data/demo-data/out-$username

# import the framework
import cmstac
import cmstaccontrib.artemis
import cmstaccontrib.rfm

# import additional packages

import cf # to read data from CF-netCDF files (https://ncas-cms.github.io/cf-python/)
import cfplot as cfp # to plot data from CF-netCDF files (https://ajheaps.github.io/cf-plot/)
from datetime import datetime, timedelta # for temporal information

# set up simulation using framework's API

td = cmstac.TimeDomain.from_start_end_step(
    start=datetime(2016, 1, 1, 0, 0, 0),
    end=datetime(2017, 1, 1, 0, 0, 0),
    step=timedelta(hours=1),
    calendar='gregorian'
)

sd = cmstac.LatLonGrid.from_extent_and_resolution(
    latitude_extent=(-90, 60),
    latitude_resolution=0.5,
    longitude_extent=(-180, 30),
    longitude_resolution=0.5
)

sd.land_sea_mask = cf.read('/data/demo-data/in/ancillary/land_sea_mask.nc')[0]
sd.flow_direction = cf.read('/data/demo-data/in/ancillary/flow_direction.nc')[0]

ds_s1 = cmstac.DataSet(
    files=['/data/demo-data/in/driving/Liddown_NFDES_CRU_v1.1.nc',
           '/data/demo-data/in/driving/Chiriqui_River_Chu_1_v1.1.nc']
)
```

16. More details about running a Jupyter Notebook.

Runs an individual cell of the notebook

One cell of the notebook. Empty brackets next to a cell means the cell has not been run yet.

Restarts a kernel and re-runs the whole notebook. Restarting a kernel will clear all output and then this button also re-runs your whole Notebook.

```
# some user-defined names
username = 'amuche'
run_id = 'my_fast_run'

# make user output directory
! [ -d /data/demo-data/out-$username ] && mkdir /data/demo-data/out-$username

# import the framework
import cmstac
import cmstaccontrib.artemis
import cmstaccontrib.rfm

# import additional packages

import cf # to read data from CF-netCDF files (https://ncas-cms.github.io/cf-python/)
import cfplot as cfp # to plot data from CF-netCDF files (https://ajheaps.github.io/cf-plot/)
from datetime import datetime, timedelta # for temporal information

# set up simulation using framework's API

td = cmstac.TimeDomain.from_start_end_step(
    start=datetime(2016, 1, 1, 0, 0, 0),
    end=datetime(2017, 1, 1, 0, 0, 0),
    step=timedelta(hours=1),
    calendar='gregorian'
)

sd = cmstac.LatLonGrid.from_extent_and_resolution(
    latitude_extent=(-90, 60),
    latitude_resolution=0.5,
    longitude_extent=(-180, 30),
    longitude_resolution=0.5
)

sd.land_sea_mask = cf.read('/data/demo-data/in/ancillary/land_sea_mask.nc')[0]
sd.flow_direction = cf.read('/data/demo-data/in/ancillary/flow_direction.nc')[0]

ds_s1 = cmstac.DataSet(
    files=['/data/demo-data/in/driving/Liddown_NFDES_CRU_v1.1.nc',
           '/data/demo-data/in/driving/Chiriqui_River_Chu_1_v1.1.nc']
)
```

You can find more commands in the Run and Kernel tabs up here. For example, commands to clear all output, re-run your Notebook etc.

After you run a cell * means that it is processing. It takes some cells longer than others do as they are performing larger tasks.

This tells you what is happening with the Notebook: Connecting means it is waiting to connect, Busy means it is processing the python commands, Idle means no command is running etc.

Once the cell finished processing, it gets a number in the brackets, which also show the order of the cells run.

If any output is written out from the cell, it is presented below the cell itself, with the same number as the cell itself. The output can be values, plots, messages etc.