THE DATA SCIENCE LAB General Introductions

COM 490 – Spring 2022

Week 1

Meet the team

Eric Bouillet
SDSC
Most modules

Sofiane Sarni
SDSC
Module 4
Week 8-10

Eloïse Doyard
Student assistant
SC-S

Cyril Matthey-Doret
SDSC
Module 1
Weeks 1-2

Olivier Verscheure
SDSC
Most modules

Haoqian Zhang
Teaching Assistant
IC IINFCOM DEDIS

John Stephan
Teaching Assistant
IC IINFCOM DCL

Baran Ozaydin
Teaching Assistant
IC IINFCOM IVRL

Dongqing Wang
Teaching Assistant
EDOC-IC

Lab Overview

- **4 Modules and 1 Final Project, 14 weeks until June 1st**
  - Module 1 - Crash course in Data Science with Python
  - Module 2 – Distributed data wrangling with Hadoop Big Data stack
  - Module 3 – Distributed machine learning with Apache Spark
  - Module 4 – Real time data acquisition and processing
  - Final – Putting it all together

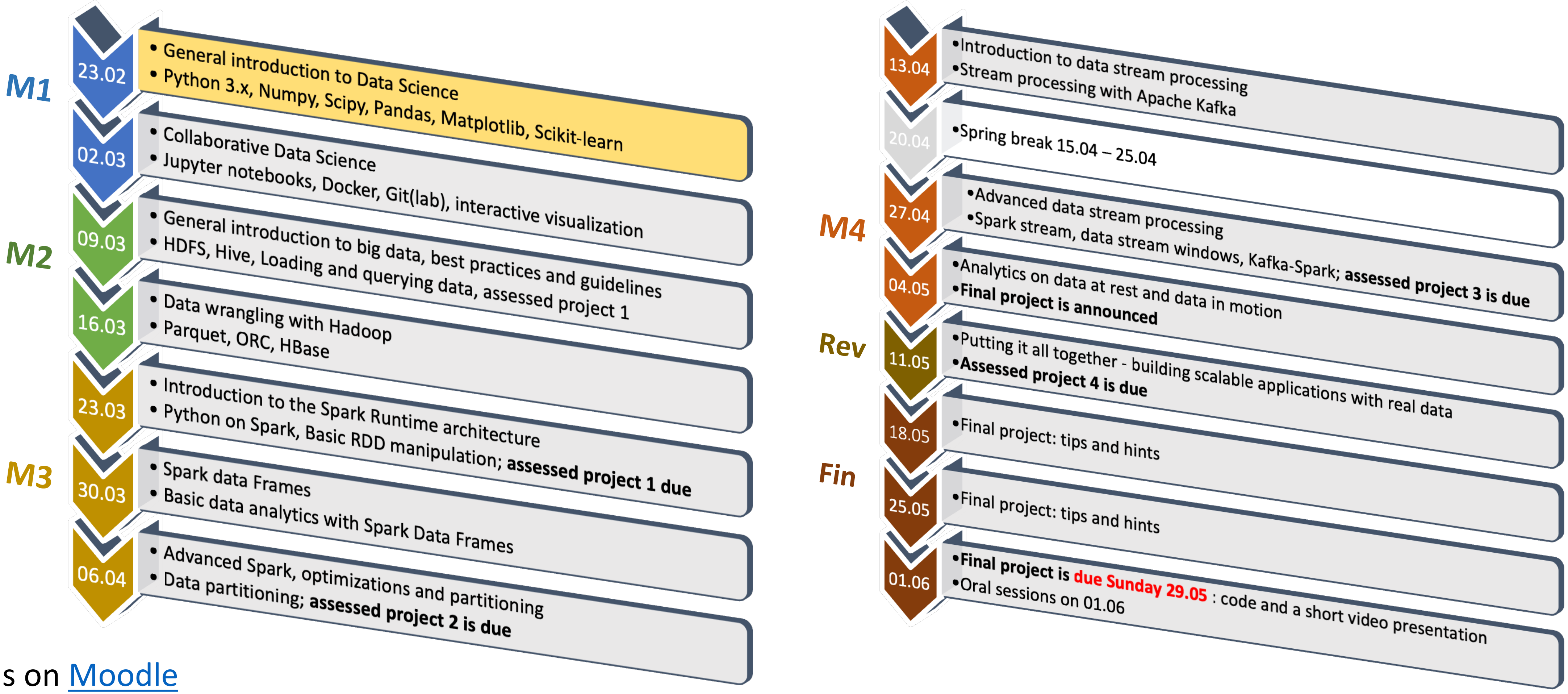
- Data Science as a journey
- Very hand-on and practical
- Team work (groups of 4)
- 3+ instructors

Lab Assessment

- 60% Continuous assessment during the semester
  - One take-home project assignment per module
  - Groups of 4 students
  - Projects due within 2 weeks (jupyter notebooks)

- 40% Final project
  - Groups of 4 students
  - Project due within 4 weeks (it will be announced with assignment 4)
  - Code, plus 6-7min video presentation
  - Mini oral (group)

Agenda Spring 2022



M1

23.02
- General introduction to Data Science
- Python 3.x, Numpy, Scipy, Pandas, Matplotlib, Scikit-learn

02.03
- Collaborative Data Science
- Jupyter notebooks, Docker, Git(lab), interactive visualization

M2

09.03
- General introduction to big data, best practices and guidelines
- HDFS, Hive, Loading and querying data, assessed project 1

16.03
- Data wrangling with Hadoop
- Parquet, ORC, HBase

M3

23.03
- Introduction to the Spark Runtime architecture
- Python on Spark, Basic RDD manipulation; **assessed project 1 due**

30.03
- Spark data Frames
- Basic data analytics with Spark Data Frames

06.04
- Advanced Spark, optimizations and partitioning
- Data partitioning; **assessed project 2 is due**

M4

13.04
- Introduction to data stream processing
- Stream processing with Apache Kafka

20.04
- Spring break 15.04 – 25.04

27.04
- Advanced data stream processing
- Spark stream, data stream windows, Kafka-Spark; **assessed project 3 is due**

04.05
- Analytics on data at rest and data in motion
- **Final project is announced**

Rev

11.05
- Putting it all together - building scalable applications with real data
- **Assessed project 4 is due**

Fin

18.05
- Final project: tips and hints

25.05
- Final project: tips and hints

01.06
- **Final project is due Sunday 29.05** : code and a short video presentation
- Oral sessions on 01.06

*Details on Moodle

Class Logistics and Format

- **Labs on Wednesday – 13h10 to 16h00**
  - First 30min to 60min - theory and general introduction to exercises
  - Class materials made available before the class on Moodle
  - 1-3 exercise sessions of 30min to 40min each, and 10min to 15min recap discussion between exercise sessions
  - Classes are recorded, and videos are made available after the class
  - Zoom https://epfl.zoom.us/j/68790504617?pwd=bHFoM29hNURLc0EzSzlUTXJ0Y3dyQT09
  - Graded assignments are due by midnight on Tuesdays before the class
  - Final is due by Sunday 29.05 midnight
- **Office hours**
  - Interactive communication via Slack workspace
  - Outside class hours on demand - time to be adapted according to students’ schedule

Agenda - Spring 2022 - Important Dates ⚠

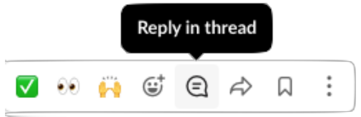
- Tuesday 22.03 – Assessment 1 is due by midnight
- Tuesday 05.04 – Assessment 2 is due by midnight
- Tuesday 26.04 – Assessment 3 is due by midnight
- Tuesday 10.05 – Assessment 4 is due by midnight
- Sunday 29.05 – Final assessment is due by midnight

- Wednesday 01.06 – Oral sessions about final project

Communication

- **Class Moodle**
  - Class schedule, announcements, class materials (slides) and other useful links
  - Updated once a week
  - https://moodle.epfl.ch/course/view.php?id=15635

- **Slack**
  - https://epfl-com490-2022.slack.com (by invitation only)
  - For real-time intra/inter group communication, and to reach us outside class hours
  - Channels:
    - #general For our general announcements or to forward EPFL guidelines
    - #meetup Looking for a team, or a teammate?
    - #random Public and unmoderated water-cooler chats
  - Etiquette:
    - **DO** Answer questions in a thread
    - **DO** Help each other with technical issues etc.,
    - **DO NOT** provide solutions to assignment

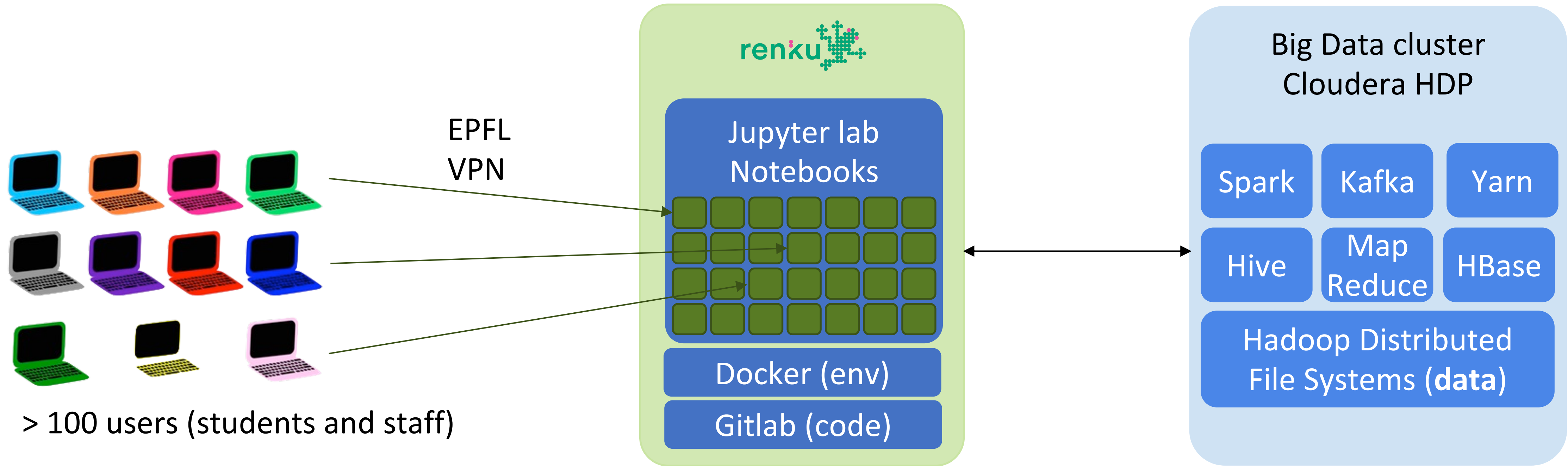


Programming Environment

- **Programming environment RENKU**
  - https://dslab2022-renku.epfl.ch (EPFL network, VPN required)
  - Login with your gaspar username and password (no need to register)

- **Programming Languages**
  - Mainly Python
    - Numpy, pandas, scikit-learn, matplotlib, PySpark (etc.) packages
  - Also SQL(-like)
  - And a pinch of Bash and Linux commands

Programming Environment



EPFL VPN

> 100 users (students and staff)

renku

Jupyter lab Notebooks

Docker (env)

Gitlab (code)

Big Data cluster Cloudera HDP

Spark

Kafka

Yarn

Hive

Map Reduce

HBase

Hadoop Distributed File Systems (**data**)

4 x IC Nodes
192 cores
1TB RAM
96TB Disk
1TB SSD

12 x IC Nodes
596 cores
3TB RAM
288TB Disk
3TB SSD

1 **BYOL**: Students work remotely using their laptops. Nothing to install – only web browser is needed.

2 Students work in teams, write and share code and environment in jupyter notebooks on Renku.

3 All data stored, and compute intensive processing executed on the distributed Hadoop Big cluster.

Programming Environment

- You will reuse the same RENKU project for all the exercises
  - The course materials for the week will be in folder ./week<n>
  - The project will be automatically updated when you start a session
- You will create a separate RENKU group project for each assignment

Today’s check list

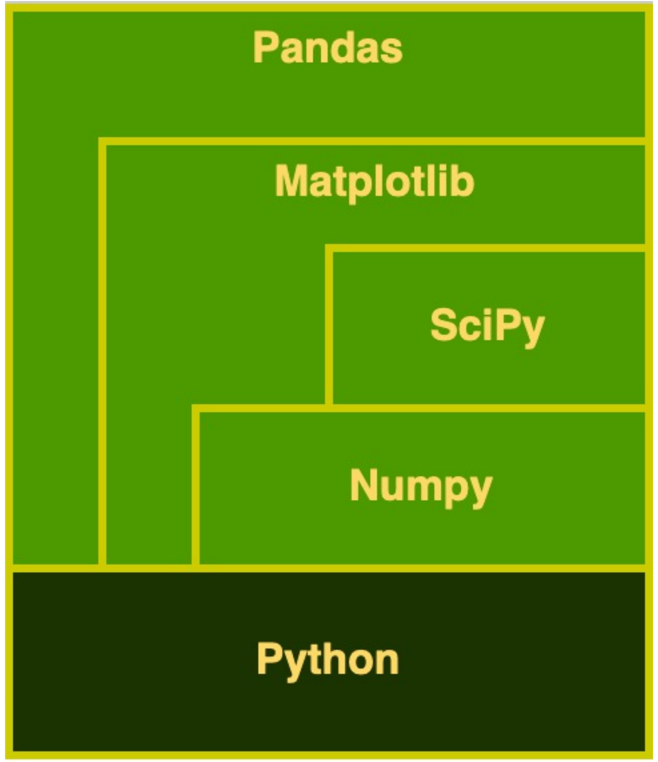
- **You have access to EPFL network (VPN)**
  - Otherwise: → https://vpn.epfl.ch
- **You have registered for the class on IS-Academia**
  - Otherwise: → http://is-academia.epfl.ch
- **You have access to our Moodle page and have bookmarked it**
  - Otherwise: → https://moodle.epfl.ch/course/view.php?id=15635
  - Contact us to add you to the list
- **You have access to our Slack workspace**
  - Otherwise: → https://epfl-com490-2022.slack.com
  - Ask us for an invite if you haven’t received one yet
- **You have access to our programming environment RENKU**
  - Otherwise: → https://dslab2022-renku.epfl.ch
  - Login with your usual EPFL (gaspar) username and password (no need to register)

General Introduction to Data Science

with Python

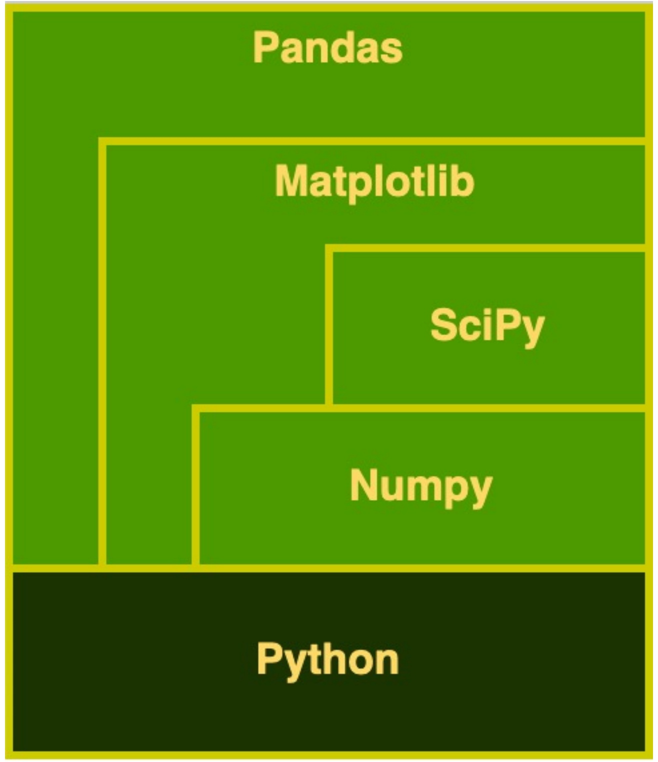
Python Mathematical Libraries

- **Python**
  - Core programming language used in the class
  - Today
    - Python types
    - List and dictionary comprehensions
    - Generators
    - Functional programming



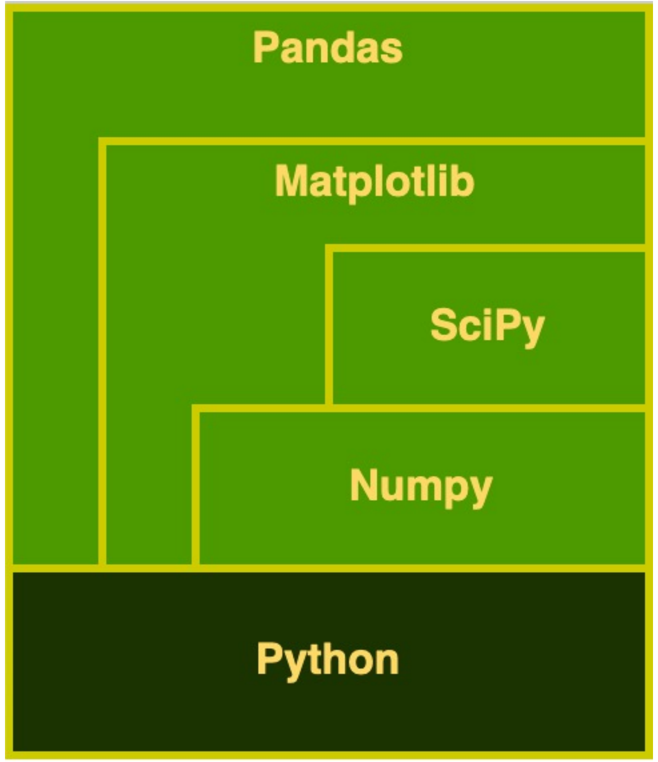
Python Mathematical Libraries

- **Numpy**
  - Core library for scientific computing in Python
  - Provides a high-performance multidimensional array object
  - Large collection of high-level mathematical functions to operate on arrays objects
  - Optimized for size and performance



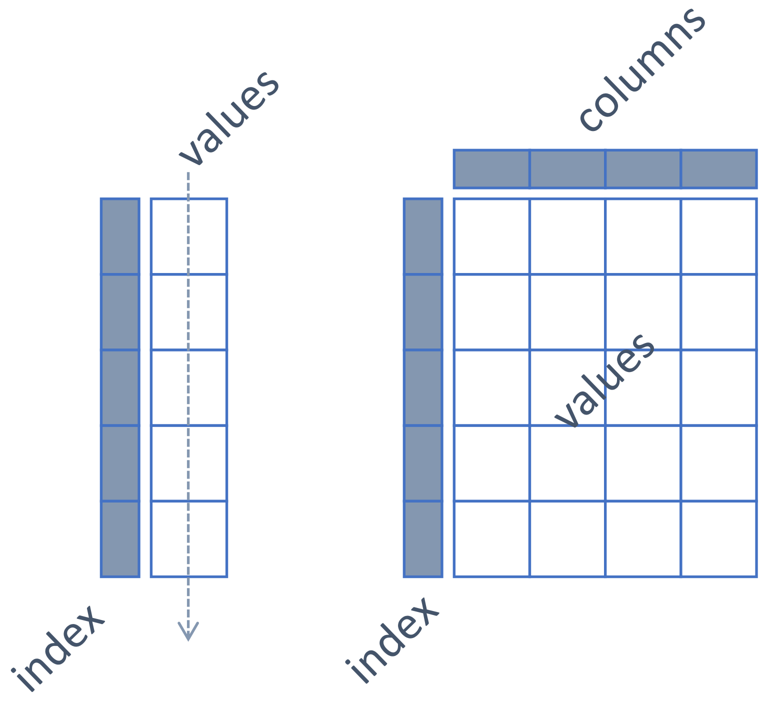
Python Mathematical Libraries

- **SciPy**
  - Built on NumPy
  - Mathematical library for Scientific and Technical Computing
    - Linear algebra, Interpolation, Integration
    - Image and signal processing, FFT
    - Linear optimization
    - Spatial algebra
    - Statistical functions
    - ...



Python Mathematical Libraries

- **Pandas**
  - Powerful & flexible data munging library
  - Built on top of NumPy
    - NumPy stores your data in arrays
    - Pandas takes the arrays, ...
      ... and gives you labelled index to it
  - Pandas data structure with columns of potentially different types
    - Basically dictionary based NumPy *ndarray*
  - Recommended reading: pandas documentation



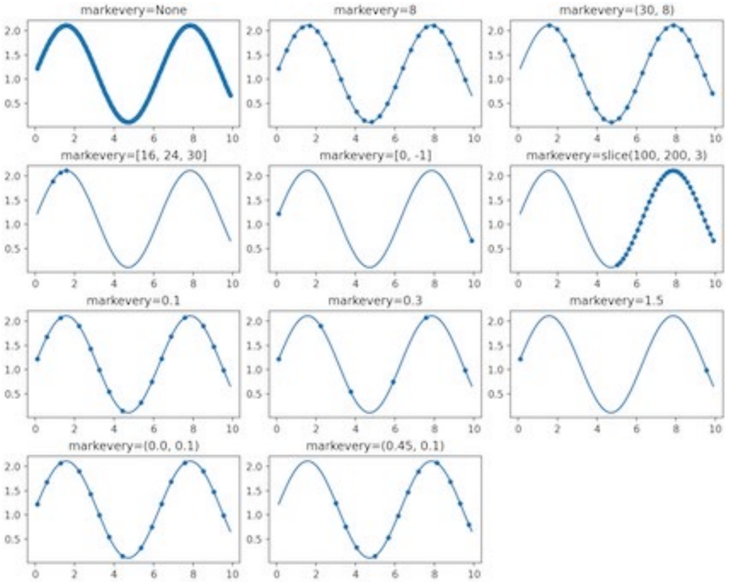
1-D
(Series)

2-D
(DataFrame)

Python Mathematical Libraries

- **Matplotlib**
  - The library for creating visualizations in Python
  - Pandas' default visualization engine

    `pandas.DataFrame.plot()`
  - Powerful, but low level programming interface
  - Best for quick and basic data exploration
- **Alternatives**
  - Plotly
  - Seaborn, folium, bokeh, osmnx, vispy, pygal, cufflinks, ...



Python Mathematical Libraries

- **Scikit-learn - Machine Learning in Python**
  - Classification, Decision Trees and Random Forests
  - Regression (logistic regression)
  - Clustering (K-Mean)
  - Nearest Neighbors
  - Dimensionality reduction (PCA)
  - Model selection (hyper-parameters)
  - ...

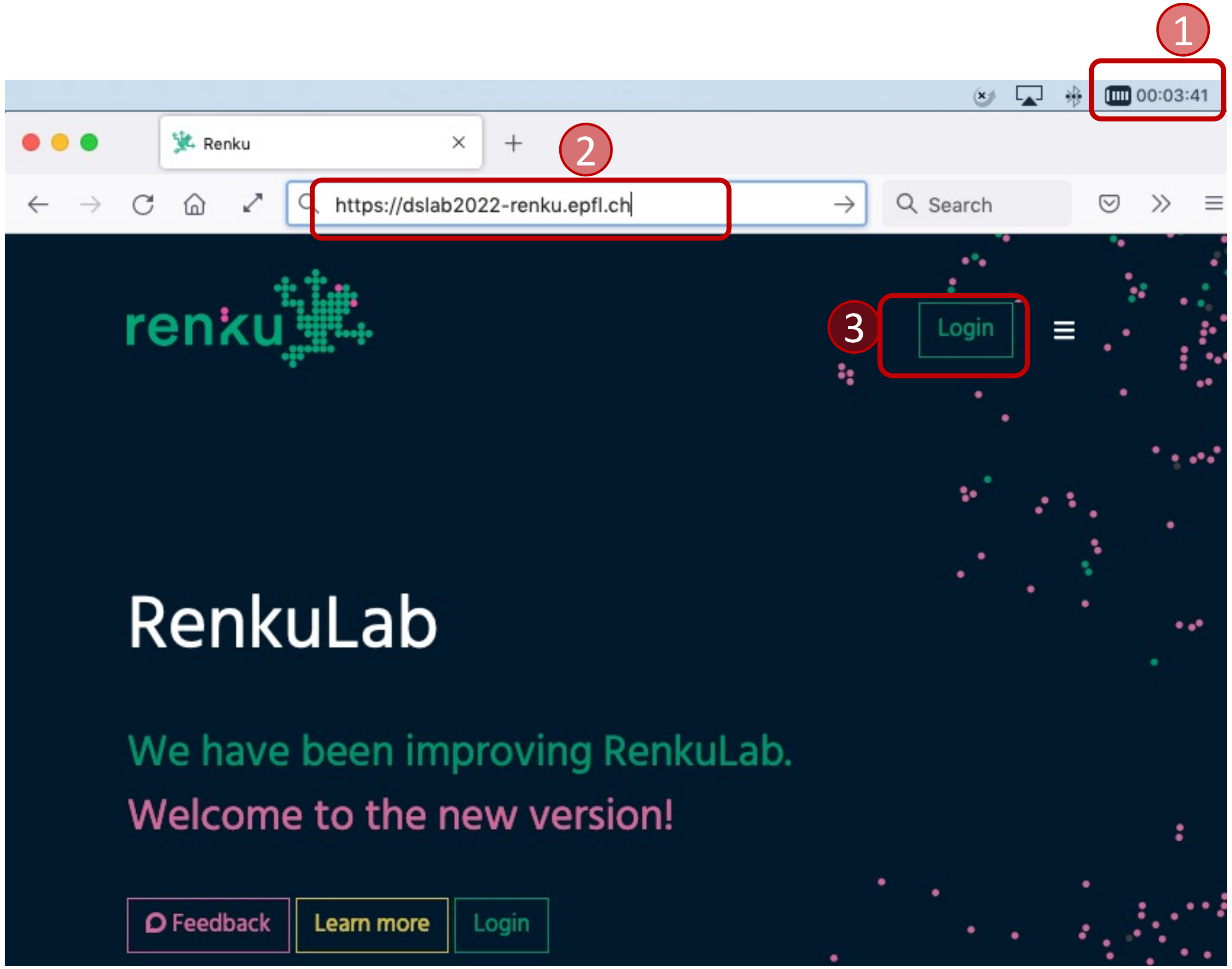
Start your engines

Bootstrapping into RENKU

https://dslab2022-renku.epfl.ch

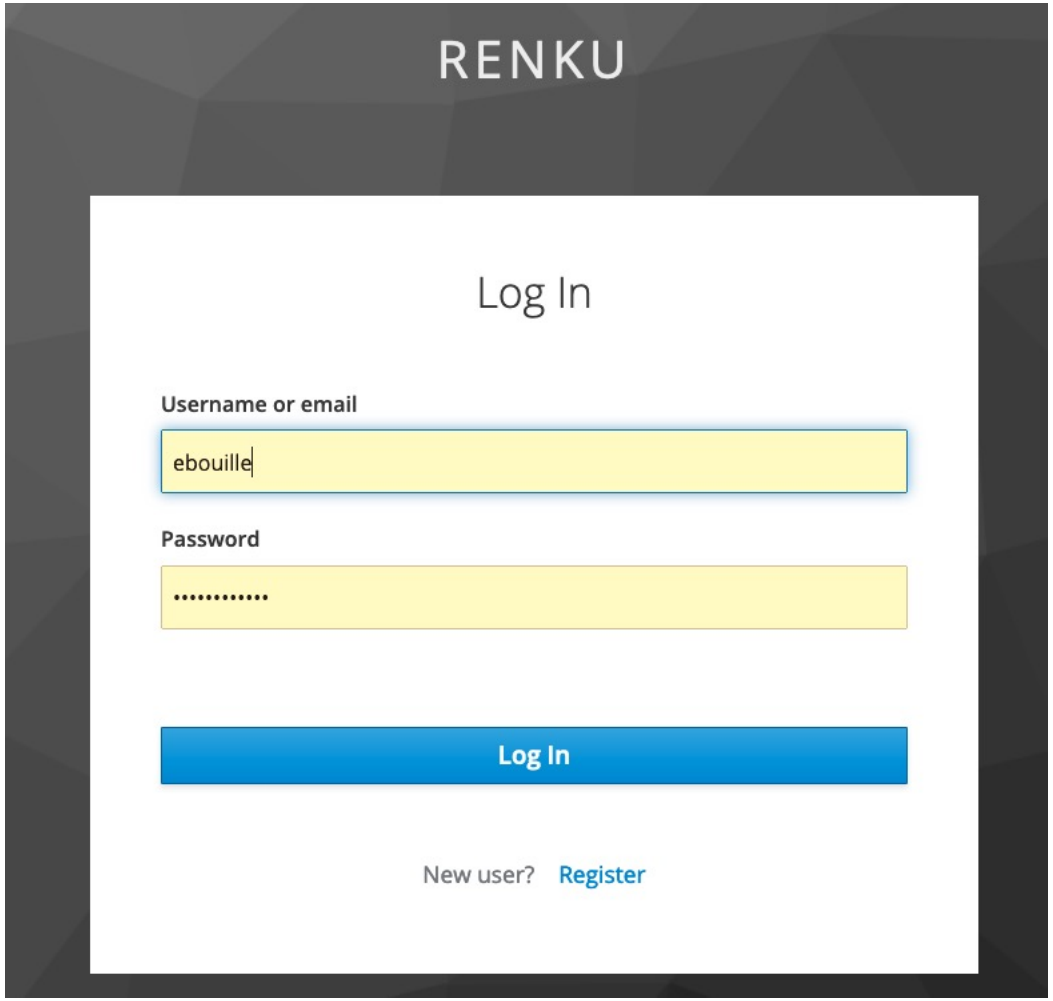
RENKU – Login

1. Must be on EPFL network (VPN)
2. Open https://dslab2022-renku.epfl.ch in a browser (Firefox, Safari, Chrome)
3. Click **Login or Sign Up**


1
00:03:41
Renku
2
https://dslab2022-renku.epfl.ch
Search
renku
3
Login
RenkuLab
We have been improving RenkuLab.
Welcome to the new version!
Feedback
Learn more
Login

RENKU – Login

- Enter your GASPAR username and Password
- Log in
- No need to register!



RENKU

Log In

Username or email

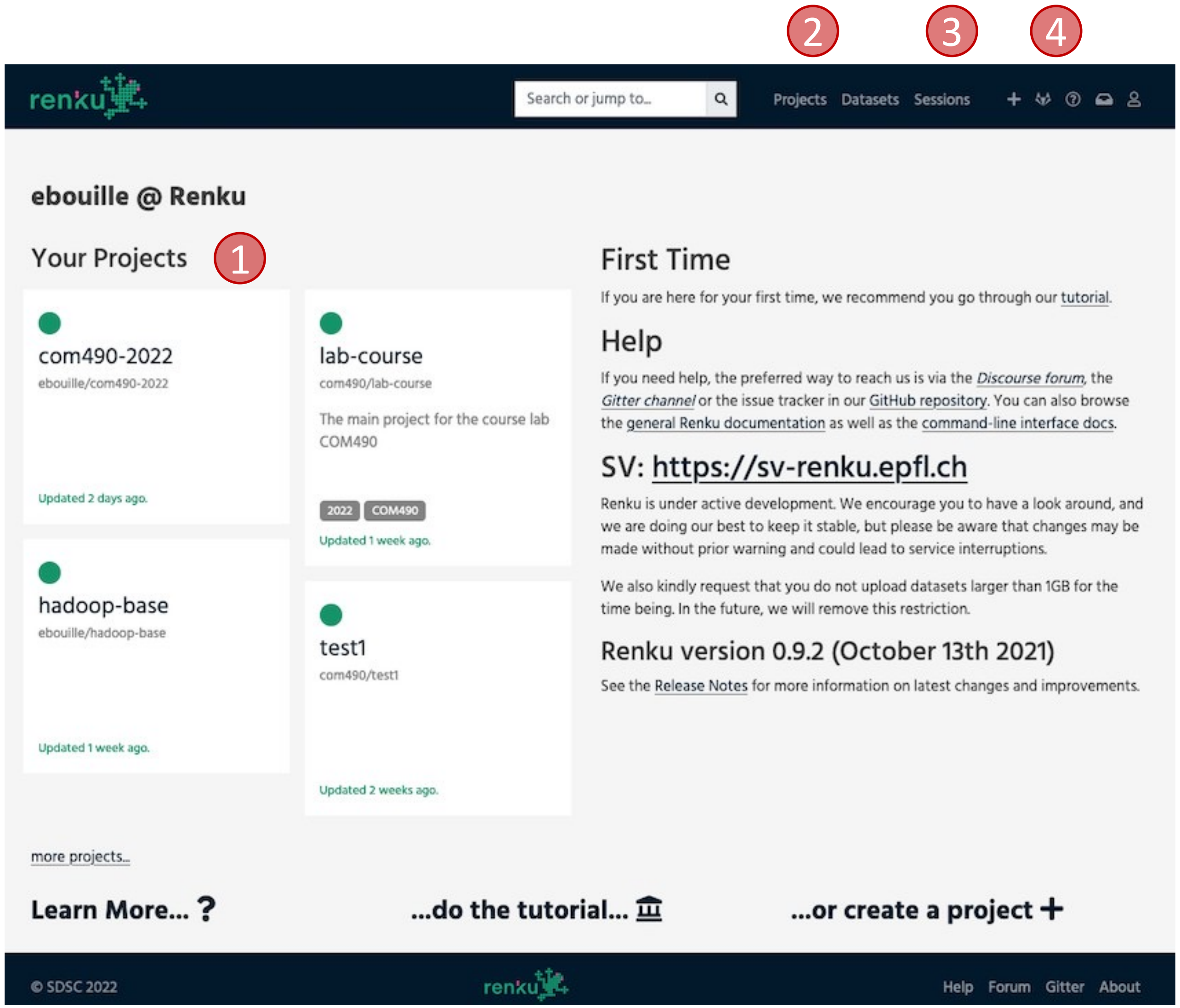
ebouille

Password

••••••••••••

RENKU - User Home Page

1. Your recent projects
2. Projects tab
3. Runtime sessions tabs
4. Gitlab view
   - Create your group
   - Manage your group members
   - Set your project visibility mode
   - and more ...



ebouille @ Renku

Your Projects

com490-2022
ebouille/com490-2022
Updated 2 days ago.

lab-course
com490/lab-course
The main project for the course lab COM490
2022 COM490
Updated 1 week ago.

hadoop-base
ebouille/hadoop-base
Updated 1 week ago.

test1
com490/test1
Updated 2 weeks ago.

more projects...

First Time

If you are here for your first time, we recommend you go through our tutorial.

Help

If you need help, the preferred way to reach us is via the *Discourse forum*, the *Gitter channel* or the issue tracker in our GitHub repository. You can also browse the general Renku documentation as well as the command-line interface docs.

SV: https://sv-renku.epfl.ch

Renku is under active development. We encourage you to have a look around, and we are doing our best to keep it stable, but please be aware that changes may be made without prior warning and could lead to service interruptions.

We also kindly request that you do not upload datasets larger than 1GB for the time being. In the future, we will remove this restriction.

Renku version 0.9.2 (October 13th 2021)

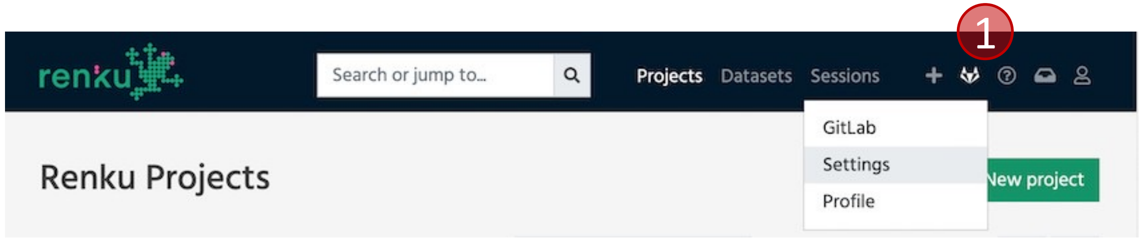
See the Release Notes for more information on latest changes and improvements.

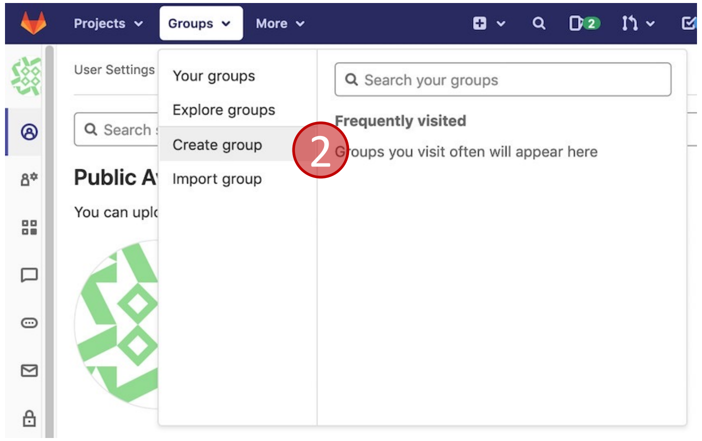
Learn More... ? **...do the tutorial...** **...or create a project +**

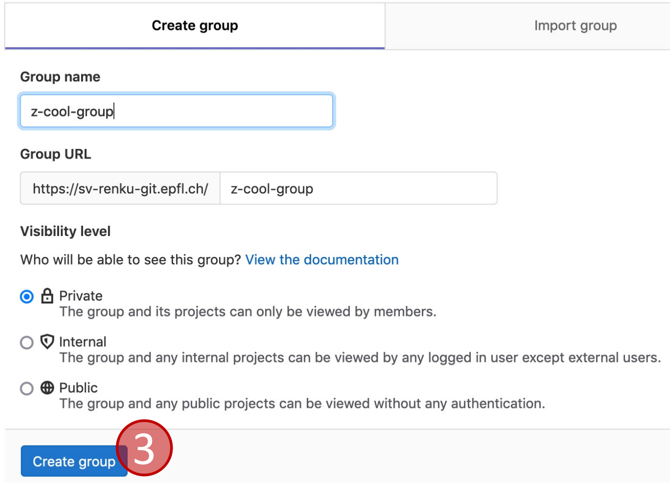
© SDSC 2022 renku Help Forum Gitter About

RENKU – Managing groups of users

1. Open any RENKU's ***Gitlab view***
2. In Gitlab, navigate to ***Groups/Create group***
3. Give a name to you group, set the visibility (private is better) and ***Create group***.
4. You can now add members to your group in the group settings (members tab)







Group name

z-cool-group

Group URL

https://sv-renku-git.epfl.ch/ z-cool-group

Visibility level

Who will be able to see this group? View the documentation

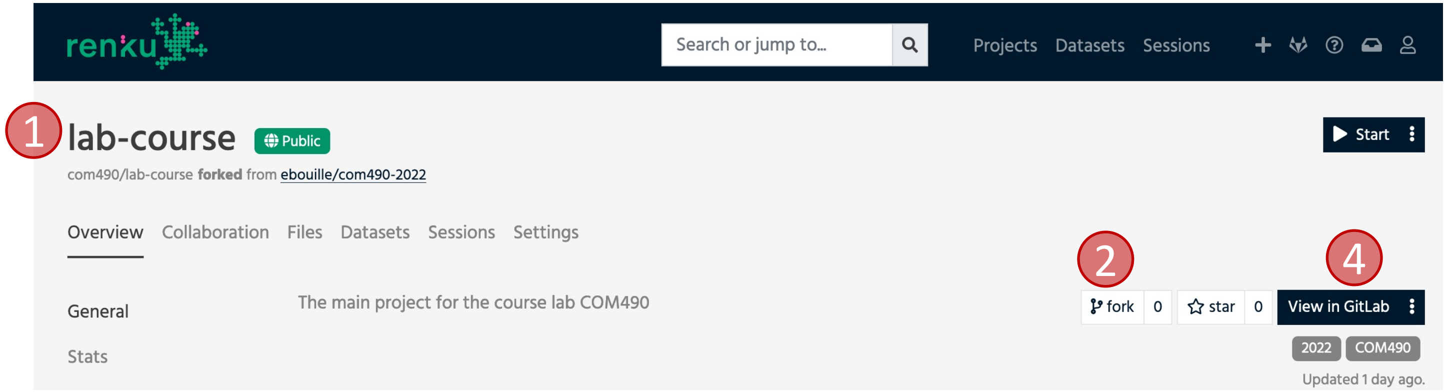
● Private
The group and its projects can only be viewed by members.

○ Internal
The group and any internal projects can be viewed by any logged in user except external users.

○ Public
The group and any public projects can be viewed without any authentication.

RENKU - Duplicating a project (fork)

1. Open the project you want to duplicate
2. Click **fork**
3. Fork under your user name, or your group name.
4. **Graded assignments and final must be set to private**
   - Open project’s ***View in Gitlab***
   - Navigate to *Settings/General/Visibility*
   - Set Visibility to ***Private***

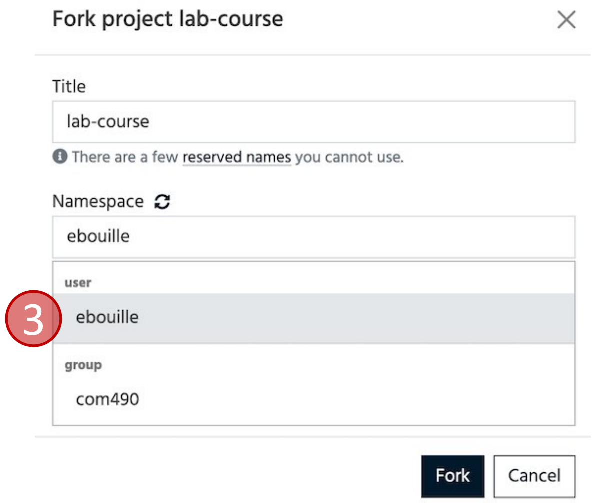


lab-course

com490/lab-course **forked** from ebouille/com490-2022

The main project for the course lab COM490

Updated 1 day ago.



Fork project lab-course

Title: lab-course

There are a few reserved names you cannot use.

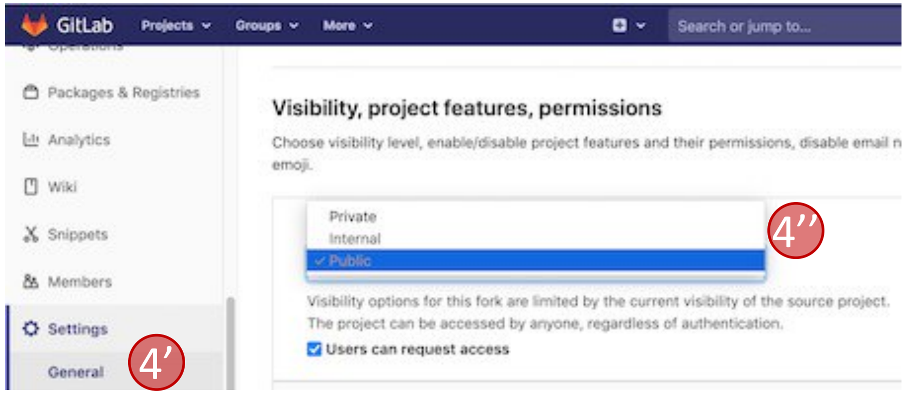
Namespace: ebouille

user

ebouille

group

com490



Visibility, project features, permissions

Choose visibility level, enable/disable project features and their permissions, disable email n emoji.

Private
Internal
✓ Public

Visibility options for this fork are limited by the current visibility of the source project.
The project can be accessed by anyone, regardless of authentication.

☑ Users can request access

RENKU – Interactive environments

Starting a new session

1. Select your project
2. Click **> Start :**

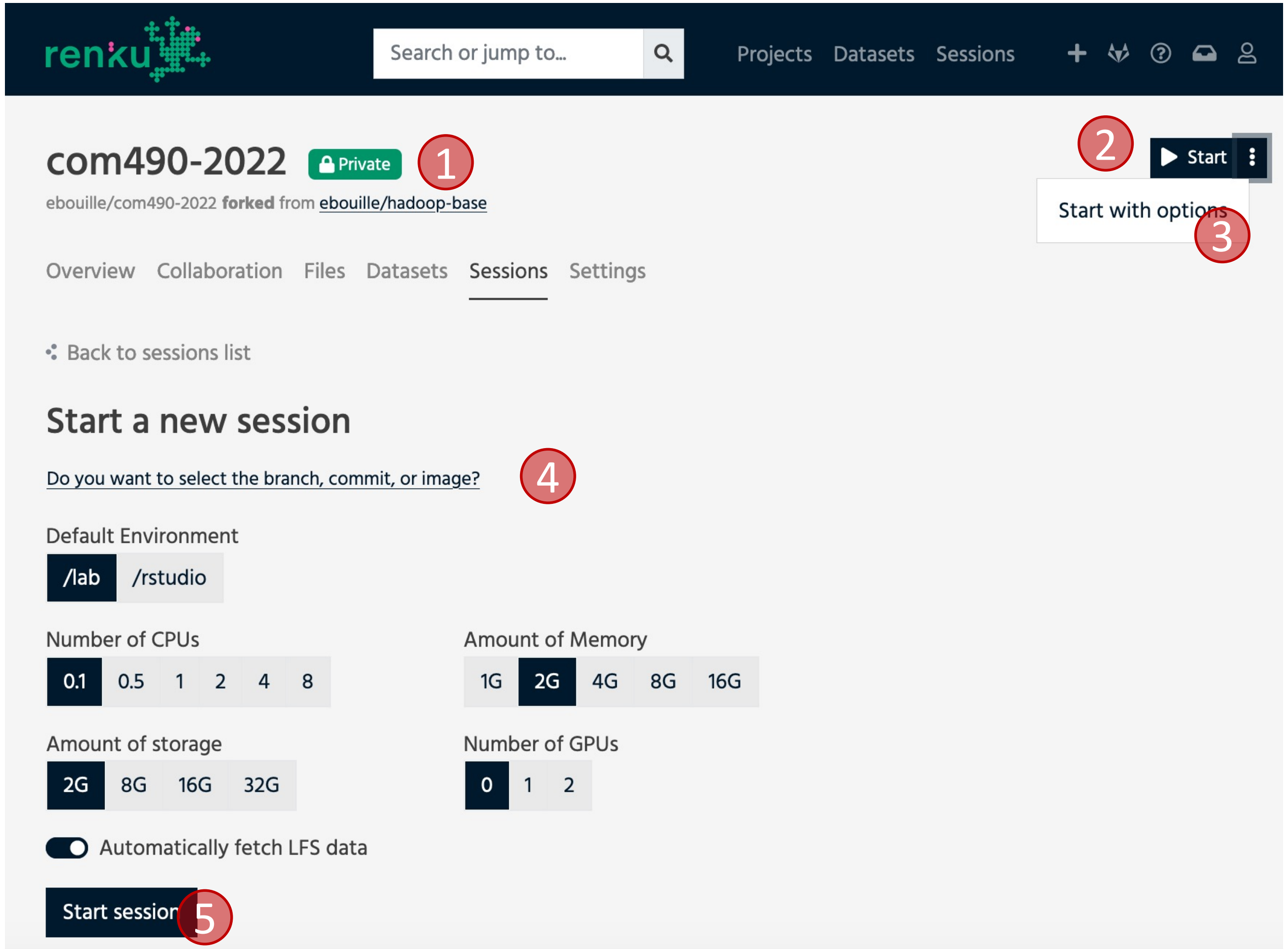
You can customize your session

3. Click **> Start :** *with options* (right of button)

   Or use the project's *Sessions/New session*

4. Configure the session
5. Click **Start session**

⚠ You may ask for more CPU and RAM. Note however that they are shared and limited resources: the more you ask, the longer you will wait until they becomes available. Furthermore that only affects the resources of your Jupyter notebook, not for the big data cluster where most of the work is done.



com490-2022

ebouille/com490-2022 **forked** from ebouille/hadoop-base

Start a new session

Do you want to select the branch, commit, or image?

Default Environment: ● /lab ○ /rstudio

Number of CPUs: ● 0.1 ○ 0.5 ○ 1 ○ 2 ○ 4 ○ 8

Amount of Memory: ○ 1G ● 2G ○ 4G ○ 8G ○ 16G

Amount of storage: ● 2G ○ 8G ○ 16G ○ 32G

Number of GPUs: ● 0 ○ 1 ○ 2

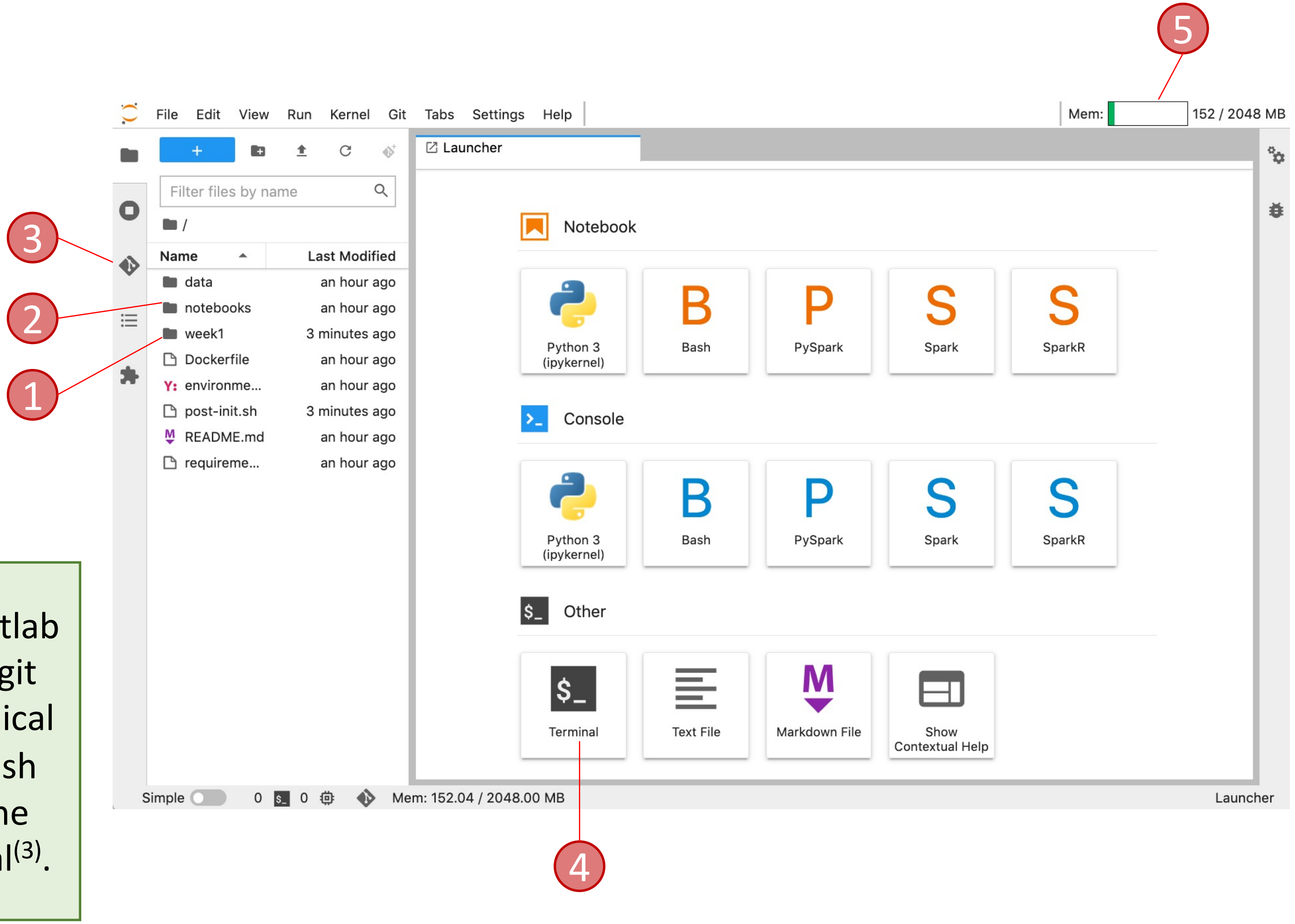
● Automatically fetch LFS data

RENKU – Working with interactive sessions

1. Folders of weekly lab

   Week1, Week2, … updated weekly

2. Folder of homework notebooks
3. Git graphical interface
4. Terminal (bash/linux)
5. Memory usage. The session should be restarted with more RAM if the gauge goes into the red.



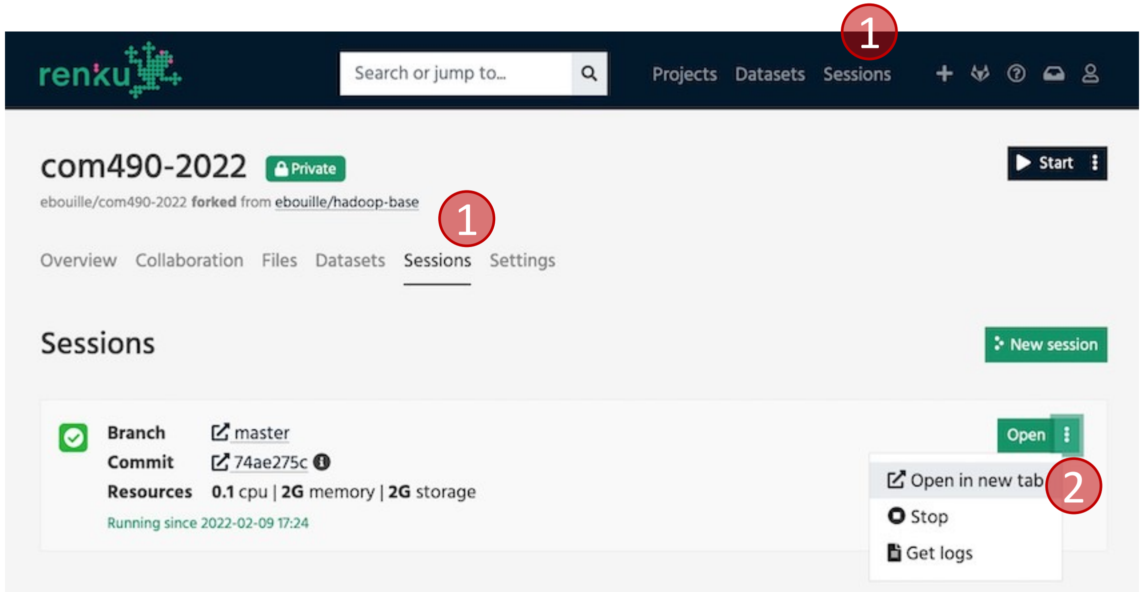
Name	Last Modified
data	an hour ago
notebooks	an hour ago
week1	3 minutes ago
Dockerfile	an hour ago
environme…	an hour ago
post-init.sh	3 minutes ago
README.md	an hour ago
requireme…	an hour ago

Saving your notebook will not save it on gitlab (it is saved locally only). You must use the git commands in the terminal or the git graphical interface in order to stage, commit and push your changes. Alternatively, you can use the **renku save** command line inside a terminal$^{(3)}$.

RENKU – Re-opening interactive sessions

Closing your browser does not stop your session immediately.
You may re-open your session at a later time as follows:

1. Select the global or the project's Sessions tab
2. **Open :** the desired session, preferably in a new browser tab



Branch master
Commit 74ae275c
Resources 0.1 cpu | 2G memory | 2G storage
Running since 2022-02-09 17:24

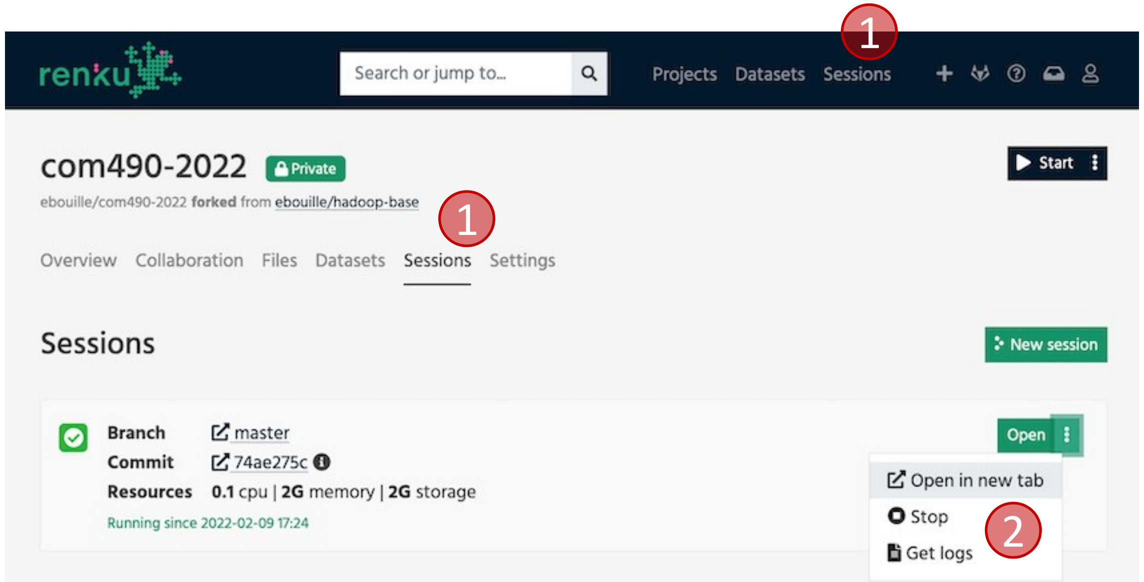
i

Your interactive environment will run for 24h after you close your browser and you can reconnect to it within 24h using this method. After that your work is git-saved on a backup branch and is stopped. By default, sessions are restarted from the backup if one is present.

RENKU – Stopping interactive sessions

Closing your browser does not stop your session immediately.
It is good etiquette to free the resources when you are done:

1. Select the global or the project's Sessions tab
2. Stop the desired session (click : on the right side of the open button)



Branch master
Commit 74ae275c
Resources 0.1 cpu | 2G memory | 2G storage
Running since 2022-02-09 17:24

⚠ Remember to save (commit and git push) your work inside your interactive environment before stopping it otherwise it will create a backup branch on gitlab.

RENKU - Etiquette

- Each user can run one interactive environment per git commit per project, that’s a lot of interactive environments
- This is a small RENKU deployment, 4 VM shared by 100 students
- Be nice to others, try to keep it to one interactive environment per person

- If you like RENKU, join us on the public https://renkulab.io instance, it's free and there is no VPN required (but no big data).