



<https://renkulab.io>

RENKU: reproducibility made easy

Renku is a tool for reproducible data science. Renku makes it possible to track your work and easily share your process and results with others.

Renkulab is our hosted platform for code environments and features for collaboration. Define a **Julia** environment and share it with others. Everyone works with the same versions of packages and tools, and it all happens at the click of a button: nothing to manage or install.

How it works

Publishing code and data alone does not make a project reproducible or even *replicable*. For this, it is necessary to provide additional information. Fortunately, there are existing tools for each of these.

What environment does the code run in?

How are code and data combined to generate results?

How did code and data evolve to the present state?

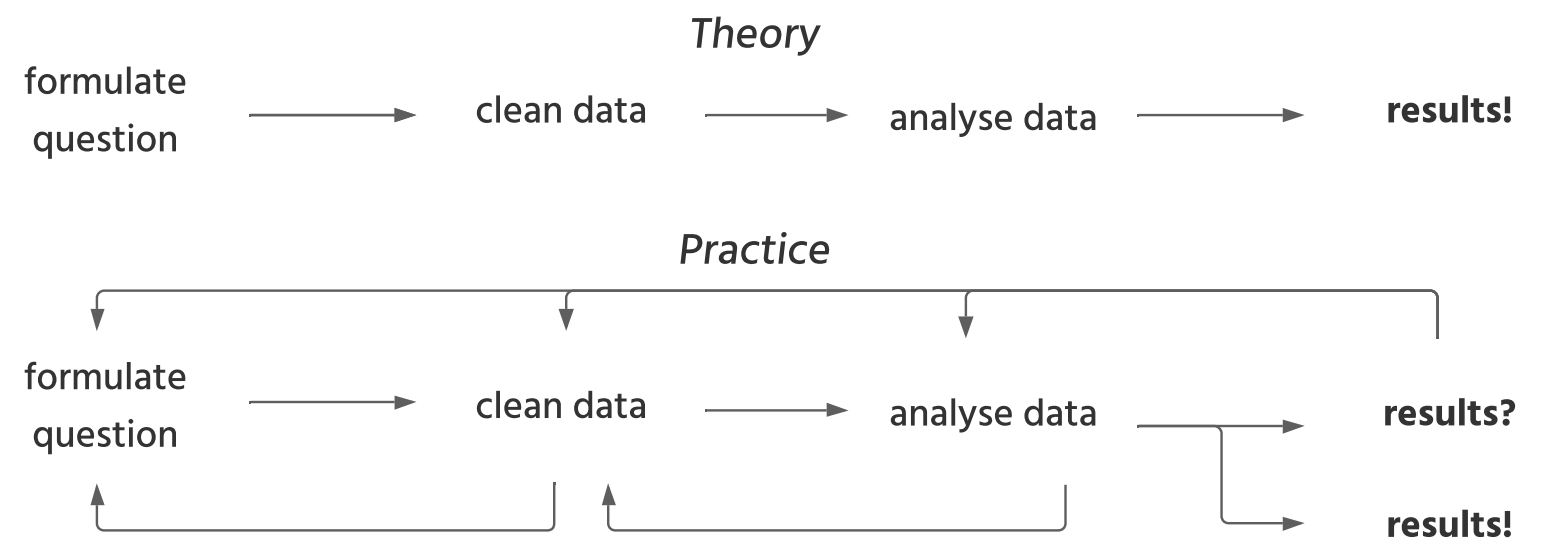
Renku bundles **git** and **Docker** together with **workflow** and **semantic web** technologies to construct a knowledge graph that binds research artifacts with their provenance and metadata, containing the information necessary to automatically replicate a result and conceptually understand how it was created.



SDSC

Swiss Data Science Center

The Data Science Process



Benefits

Everyone benefits from reproducible processes, but the main beneficiary is **you**, yourself.

In theory, the data science process is straightforward and linear.

In practice, it is messy. As you work, you backtrack, revise your ideas and update your approach.

With Renku, your work is tracked and reproducible, giving you the confidence to try things out and explore avenues. You can always return to the last working state if necessary. And months later, you will still know *why* you did what you did.

We support Julia

With Renkulab, you can work in **Julia**. Your colleagues can view your results and step through your analysis and execute your code, even if they are not Julia developers themselves.

We provide a ready-to-go Julia template for Renku projects with no additional set up required.

Getting start on Renku with Julia

Creating a Julia project in Renkulab is easy.

Our base Julia template works out of the box, but you can customise it if you are a power user.

Projects Datasets Environments

New project

Title

There are a few reserved names you cannot use.

Namespace ↻

Identifier

This is automatically derived from Namespace and Title.

Visibility

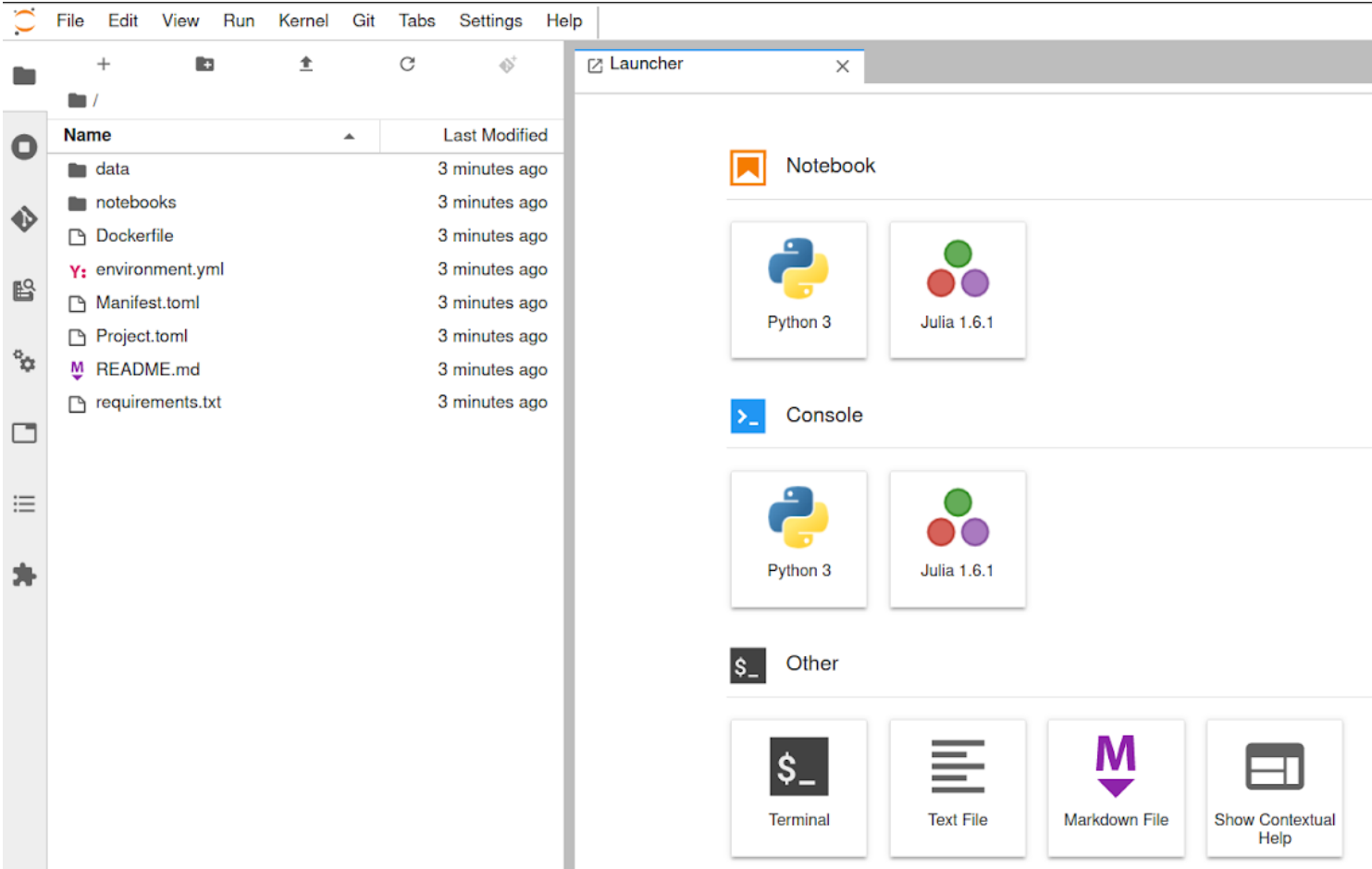
Template source

Template

The simplest Julia 1.6.1-based renku project with a basic directory structure and necessary supporting files.

Working with Julia on Renkulab

Once you have configured and started your session, you can access a host of tools from the JupyterLab interface to help you do data science with Julia. Jupyter Notebooks with Julia kernels are available to prototype your work, whilst the terminal allows you to access the Renku command-line interface, which is installed as default on Renku projects.



Configuring your session

On renkulab.io, we provide free computational resources to get you up and running.

Define your configuration and start your session with one click.

Start a new interactive environment

Branch (only 1 available) ↻ ⚙️

Commit ↻ ⚙️

Docker Image available

Default Environment

Number of CPUs

Amount of Memory

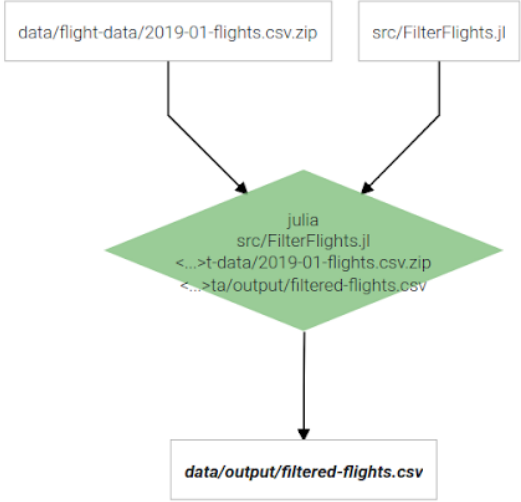
Number of GPUs: 0

Automatically fetch LFS data

Renku commands

Your workflows can be tracked and made reproducible by running them through *renku run*. Metadata is generated behind the scenes and you can view it graphically on your project page.

```
Terminal 3
base > work > my-first-renku-project-with-julia > master > $ renku run julia src/FilterFlights.jl \
> data/flight-data/2019-01-flights.csv.zip \
> data/output/filtered-flights.csv
Reading data/flight-data/2019-01-flights.csv.zip ...
Writing data/output/filtered-flights.csv ...
Info: Adding these files to Git LFS:
data/output/filtered-flights.csv
To disable this message in the future, run:
renku config set show_lfs_message False
```



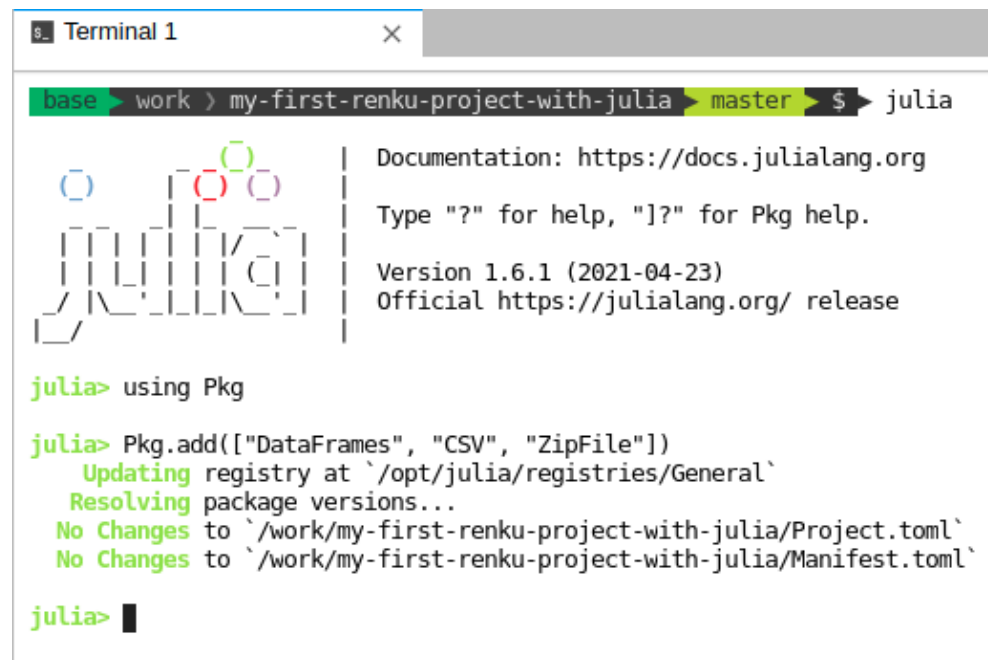
Extending Renku: for Julia and beyond

Renku is designed to integrate into your existing stack(s) and work habits, not compete with them. If you are an advanced user, you can customise your environment and experience using Renkulab.

Managing packages

The JupyterLab Terminal allows you to access the Julia command line directly. From there, the familiar **Pkg** manager can be used to add dependencies.

The Project.toml and Manifest.toml file, which come as defaults from the standard Julia template, will be updated accordingly.



```
Terminal 1
base work > my-first-renku-project-with-julia > master > $ julia
Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.
Version 1.6.1 (2021-04-23)
Official https://julialang.org/ release

julia> using Pkg

julia> Pkg.add(["DataFrames", "CSV", "ZipFile"])
Updating registry at `~/opt/julia/registries/General`
Resolving package versions...
No Changes to `~/work/my-first-renku-project-with-julia/Project.toml`
No Changes to `~/work/my-first-renku-project-with-julia/Manifest.toml`

julia> █
```

Customising your container

We use **Docker** to manage the container from which your session loads. Each project has its own **Dockerfile** which can be customised to add any Linux software you may need as part of your project. You can also set up environment variables there which will be defined as soon as your session is launched.

Need the same defaults all the time? Make a template!

For advanced users, or those teaching a course, there are often a list of dependencies or files needed for a large group of users. In this case, you can define a Renku project template which can be shared and instantiated across users.

We maintain a stack of them here:

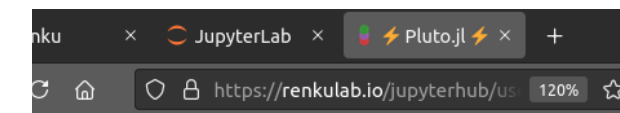
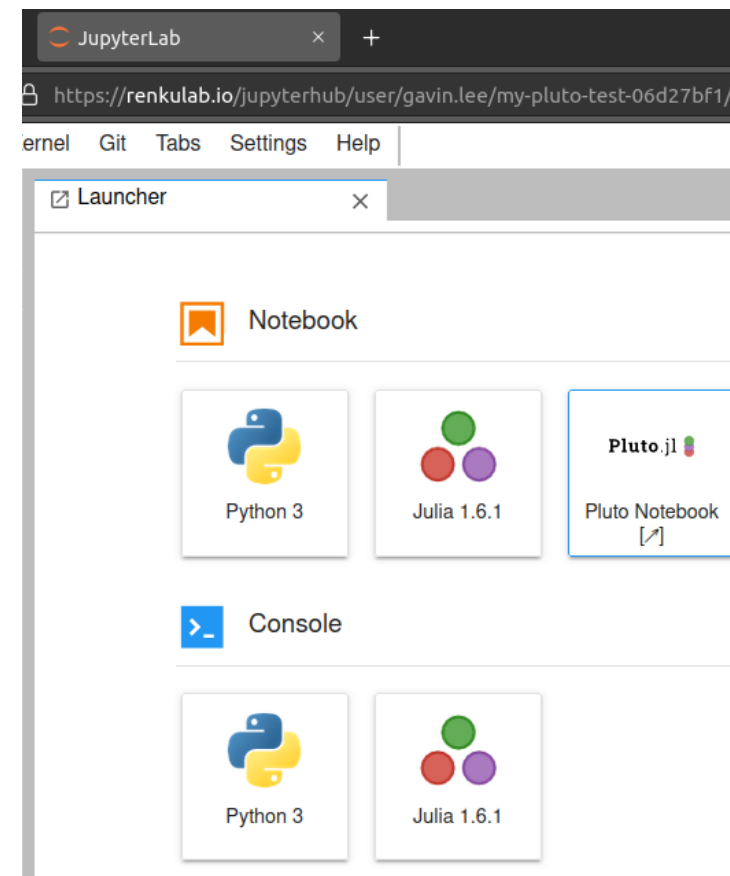
<https://github.com/SwissDataScienceCenter/contributed-project-templates>, and we encourage users to define templates that may be useful for their own Julia community.

Pluto.jl

Renku's sessions are built up from JupyterHub, which itself allows a host of extensions. We are able to serve other end-points, like dashboards and virtual desktops.

Using such extensions is possible to extend Renku to be able to host, for example, Pluto notebooks, which are a popular alternative to Jupyter Notebooks themselves. They are plain .jl files and allows for smart re-execution of dependent cells.

You can then fire up a Pluto notebook and a new tab will take you to Pluto.jl! The Jupyter Notebook option will remain as a default, so Renku can be used flexibly.



welcome to Pluto.jl

- Samples :
1. [Getting started](#)
 2. [Basic mathematics](#)
 3. [Interactivity](#)
 4. [Pluto.jl](#)

Get in touch



Ask questions @ Discourse

<https://renku.discourse.group/>



Chat about Renku @ Gitter

<https://gitter.im/SwissDataScienceCenter/renku>



Check out manuals & tutorials

https://renku.readthedocs.io/en/latest/getting_started.html



Request features & report bugs @ GitHub

<https://github.com/SwissDataScienceCenter/renku>

Authors

Gavin Lee

gavin.lee@datascience.ch

Chandrasekhar Ramakrishnan

cramakri@ethz.ch

Rok Roškar

roskarr@ethz.ch

Oksana Riba Grognez

Ralf Grubenmann

Tao Sun

Mohammad Alisafee

Christine Choirat

Viktor Gal

Andreas Bleuler

Lorenzo Cavazzi

Pamela Delgado

Virginia Friedrich

Tasko Olevski

Jakub Chrobasik

Alessandro Degano

Travis Lee

David Kunzmann

