Community Hydrological Modelling Workflows

Online Training Session with Federal, Provincial, and Territorial Partners

Kasra Keshavarz
Research Scientist
Schulich School of Engineering
University of Calgary

Alain Pietroniro, PhD, PEng
Professor and Canada Research Chair
Schulich School of Engineering
University of Calgary

March 21st, 2024

UNIVERSITY OF CALGARY

Acknowledgements

GLOBAL WATER FUTURES

Ouranos

Environment and Climate Change Canada

Environnement et Changement climatique Canada

Alberta Government

Digital Research Alliance of Canada | Alliance de recherche numérique du Canada

Yukon

Water Security Agency

Special thanks to:

- Wouter Knoben (Ucalgary), Shervan Gharari (Usask), Martyn Clark (Ucalgary), Ala Bahrami (Usask), Guoqiang Tang (NCAR), Cooper Albano (Usask), John Pomeroy (Usask), Dan Princz (ECCC), Travis Logan (Ouranos), Pierre Pellerin (Ouranos), Dikraa Khedhaouiria (ECCC) and everyone present today in the session.

UNIVERSITY OF CALGARY

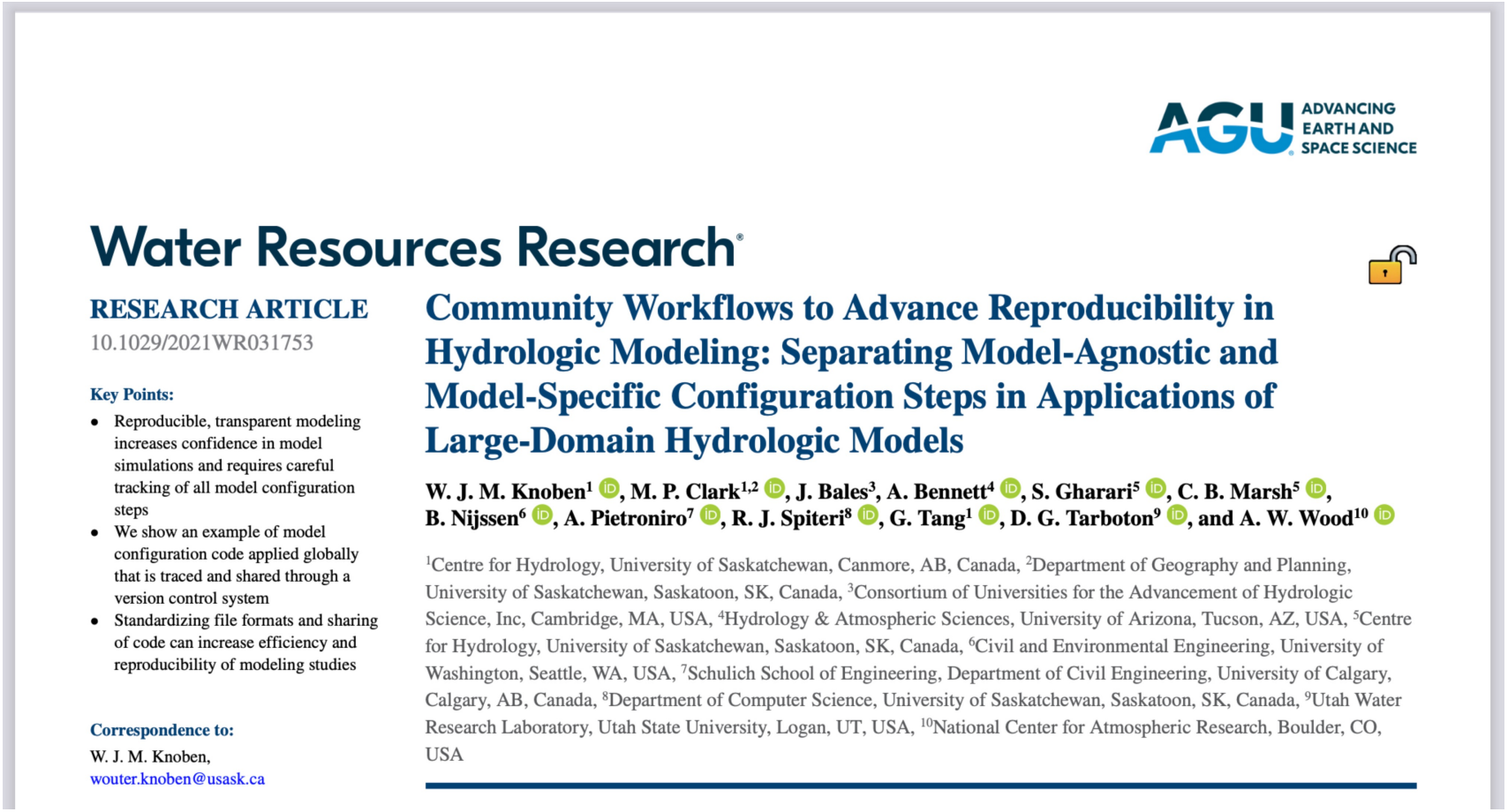
Outline

- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Outline

- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Recipe for Community Hydrological Modelling



AGU ADVANCING EARTH AND SPACE SCIENCE

Water Resources Research

RESEARCH ARTICLE
10.1029/2021WR031753

Key Points:

- Reproducible, transparent modeling increases confidence in model simulations and requires careful tracking of all model configuration steps
- We show an example of model configuration code applied globally that is traced and shared through a version control system
- Standardizing file formats and sharing of code can increase efficiency and reproducibility of modeling studies

Correspondence to:
W. J. M. Knoben,
wouter.knoben@usask.ca

Community Workflows to Advance Reproducibility in Hydrologic Modeling: Separating Model-Agnostic and Model-Specific Configuration Steps in Applications of Large-Domain Hydrologic Models

W. J. M. Knoben[1], M. P. Clark[1,2], J. Bales[3], A. Bennett[4], S. Gharari[5], C. B. Marsh[5], B. Nijssen[6], A. Pietroniro[7], R. J. Spiteri[8], G. Tang[1], D. G. Tarboton[9], and A. W. Wood[10]

[1]Centre for Hydrology, University of Saskatchewan, Canmore, AB, Canada, [2]Department of Geography and Planning, University of Saskatchewan, Saskatoon, SK, Canada, [3]Consortium of Universities for the Advancement of Hydrologic Science, Inc, Cambridge, MA, USA, [4]Hydrology & Atmospheric Sciences, University of Arizona, Tucson, AZ, USA, [5]Centre for Hydrology, University of Saskatchewan, Saskatoon, SK, Canada, [6]Civil and Environmental Engineering, University of Washington, Seattle, WA, USA, [7]Schulich School of Engineering, Department of Civil Engineering, University of Calgary, Calgary, AB, Canada, [8]Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, [9]Utah Water Research Laboratory, Utah State University, Logan, UT, USA, [10]National Center for Atmospheric Research, Boulder, CO, USA

Knoben, W. J. M., Clark, M. P., Bales, J., Bennett, A., Gharari, S., Marsh, C. B., Nijssen, B., Pietroniro, A., Spiteri, R. J., Tang, G., Tarboton D. G. & Wood, A. W. (2022). Community Workflows to Advance Reproducibility in Hydrologic Modeling: Separating Model-Agnostic and Model-Specific Configuration Steps in Applications of Large-Domain Hydrologic Models. *Water Resources Research*, *58*(11), e2021WR031753.

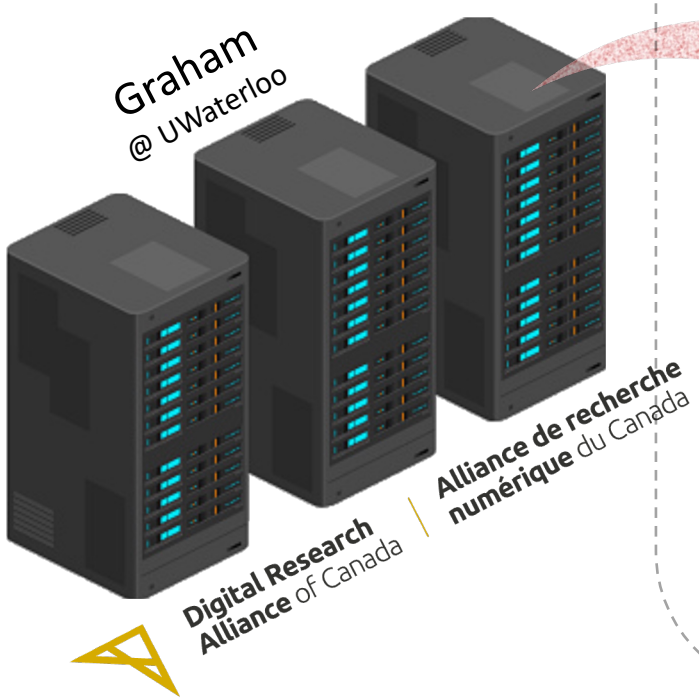
- Developing relevant workflows on High Performance Computing (HPC) facilities
- Keeping workflows open source and reproducible for the scientific community
- Keeping workflows efficient for real-world applications
- Separate “model-agnostic” and “model-specific” configuration parts

UNIVERSITY OF CALGARY

Outline

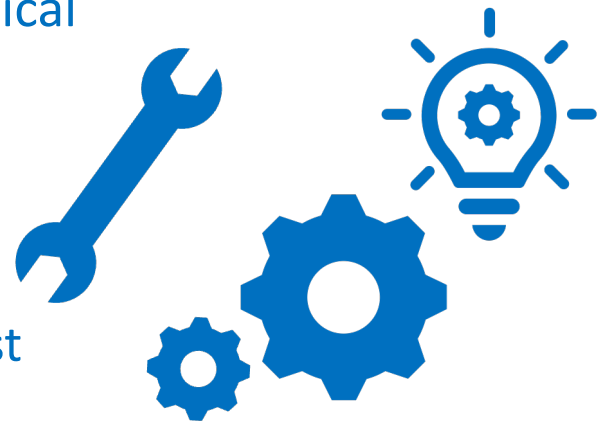
- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Digital Infrastructure



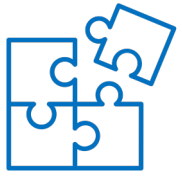
Suite of tools to facilitate hydrological modelling and data processing:

- Generalizing approaches applicable to data processing applications
- Using stable tools to assure robust performance on HPC platforms




HPC Tailored


Visualize

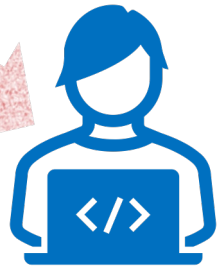

Regrid


Subset


Temporal Disaggregate


Remap

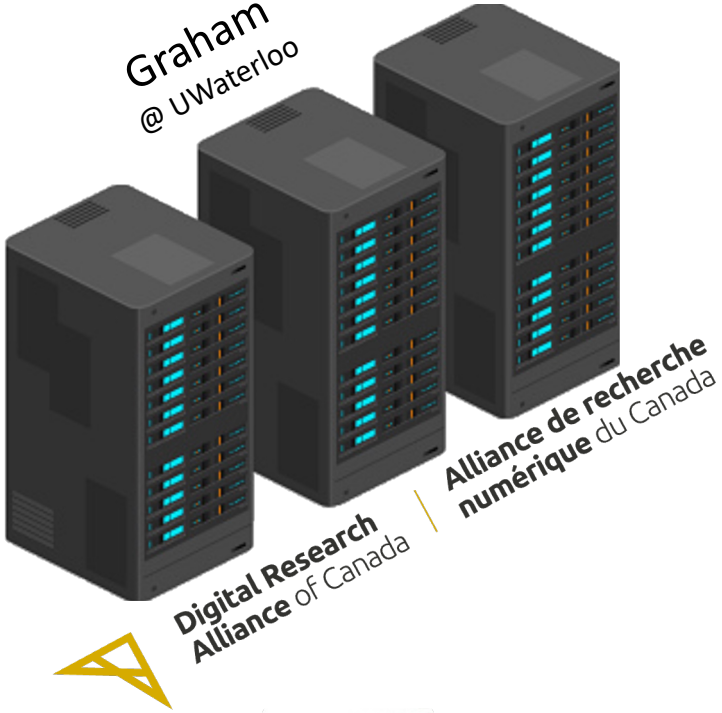
Science and Policy Decisions



Credit: Dr. Shervan Gharari

UNIVERSITY OF CALGARY

Digital Infrastructure



Remote access accounts

`rpp-kshook`

`rrg-mclark`

UNIVERSITY OF CALGARY

Digital Infrastructure

Our recommendation is to access Graham using either of the following methods:

jupyterlab

https://jupyterhub.sharcnet.ca/

- Multi-user environment
- Resource efficient
- Collaborative environment
- Reproducibility mechanisms
- Strong community support
- Ease of use

Terminal

```
$ ssh USERNAME@graham.alliancecan.ca
```

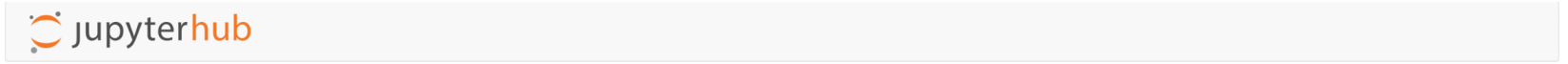
- Maximal flexibility
- Remote Access
- Steep learning curve
- Full control of command sessions
- Strong community support

MobaXTerm PuTTY MacOS Terminal

UNIVERSITY OF CALGARY

Any questions so far?

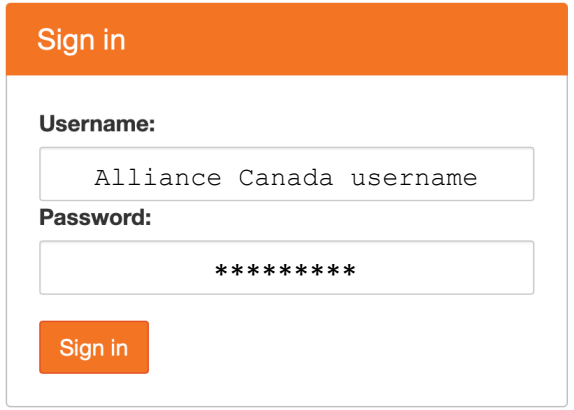
Digital Infrastructure



https://jupyterhub.sharcnet.ca/

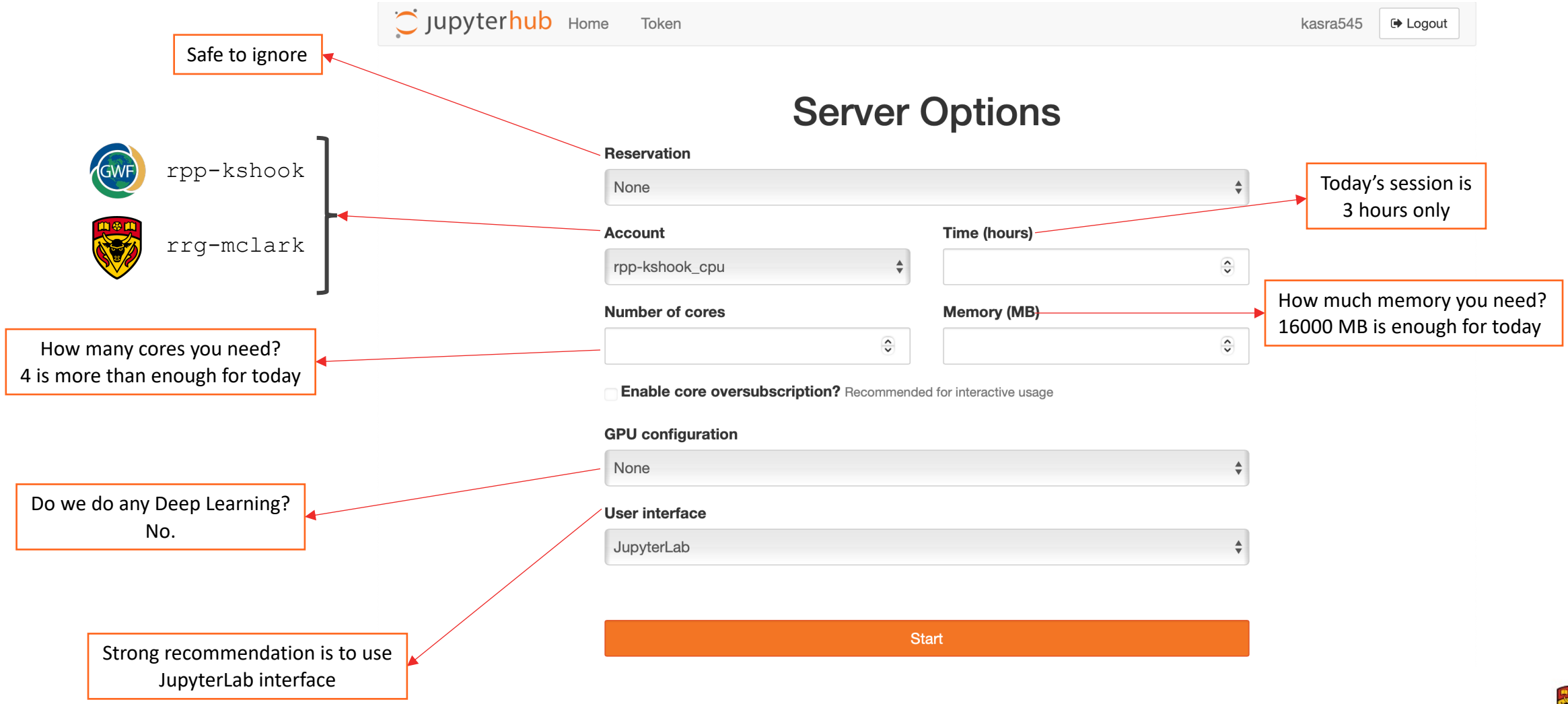
jupyterhub

- Multi-user Server Service for Jupyter Sessions
- User Authentication Capability
- Resource Customization
- Customizable Work Environment



UNIVERSITY OF CALGARY

Digital Infrastructure



UNIVERSITY OF CALGARY

Digital Infrastructure

https://jupyterhub.sharcnet.ca/

jupyterhub

DALLE generated image

General tips:

- Understand your setup: Be resourceful! Asking too much computational resource can keep you in the queue forever; public HPCs are shared resources. Asking for too little resources can make your Jupyter session slow and laggy.

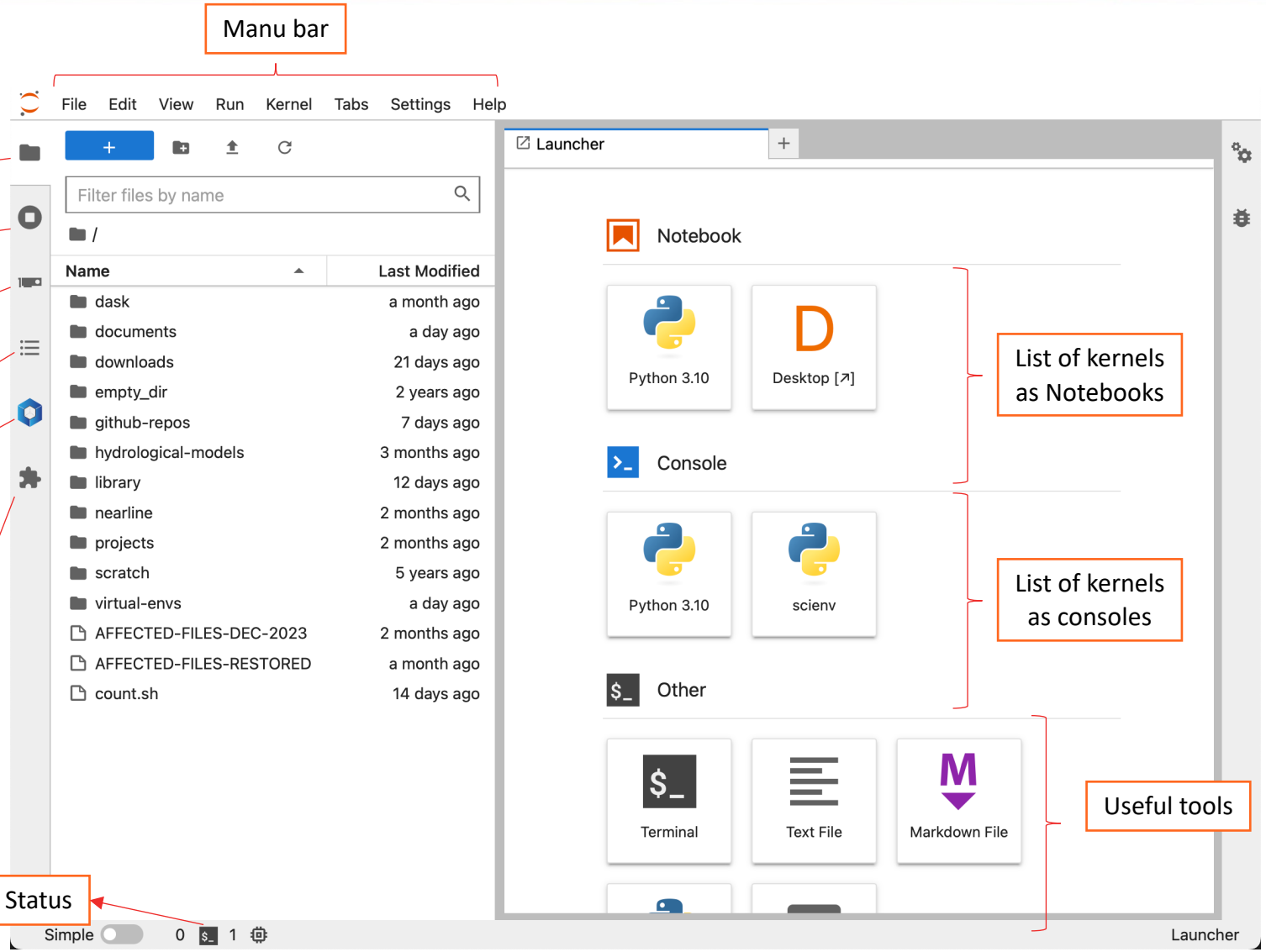
DALLE generated image

Choose the Right Kernel: Be mindful! Jupyter provides you with all sort of programming environments; make sure to choose the right (customized) programming kernel for your work.

Monitor Performance and Document: Be diligent! As you work more with the environment, you get to know the requirements of your computations. Keep track of your workflow's performance and document your work.

UNIVERSITY OF CALGARY

Digital Infrastructure



UNIVERSITY OF CALGARY

Digital Infrastructure

HPCs comes with various “**modules**”

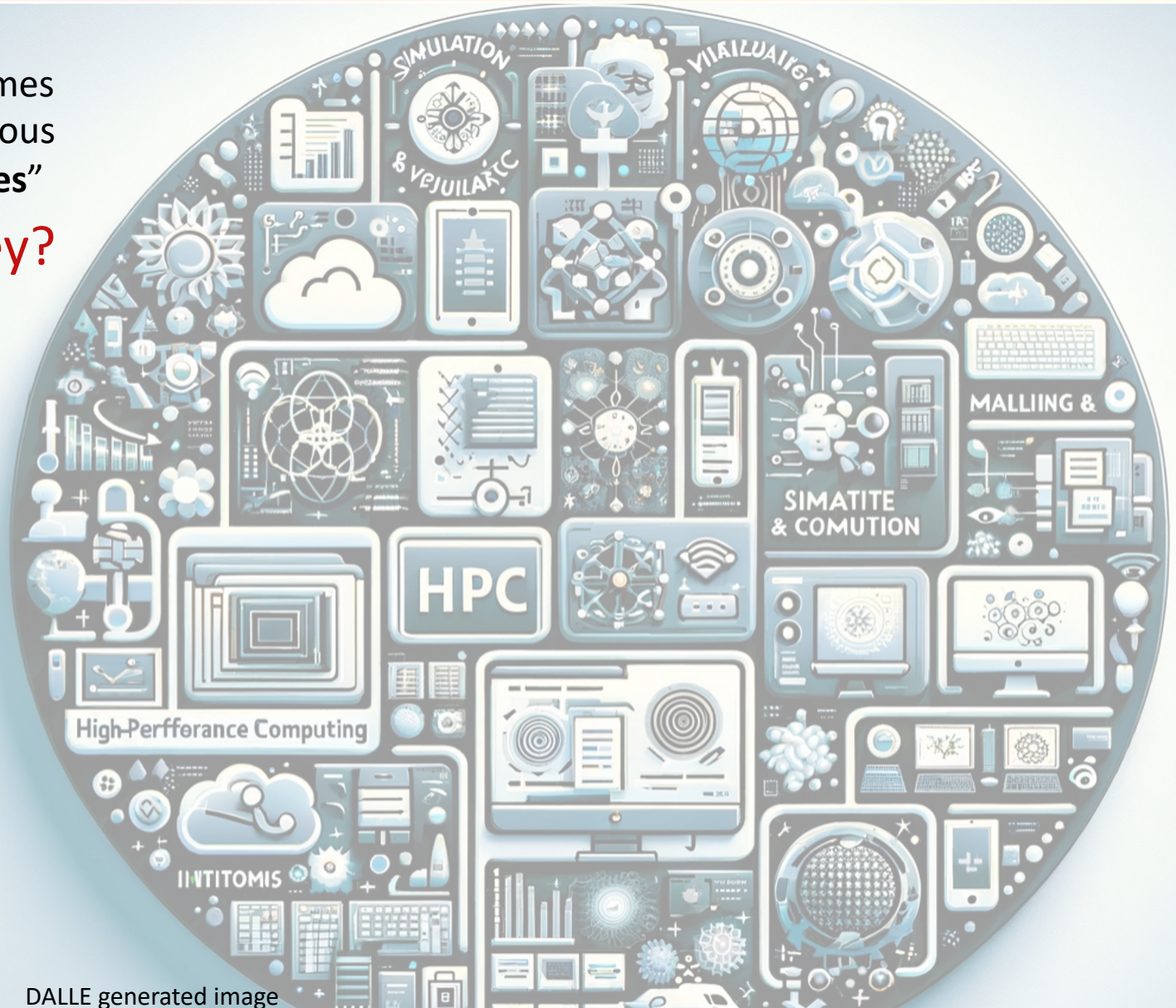
But where are they?

Flexibility

Different users have different requirements

Customization

Users may need specific versions of software or libraries



DALLE generated image

Optimization

software packages are often optimized for the underlying hardware architecture

Productivity

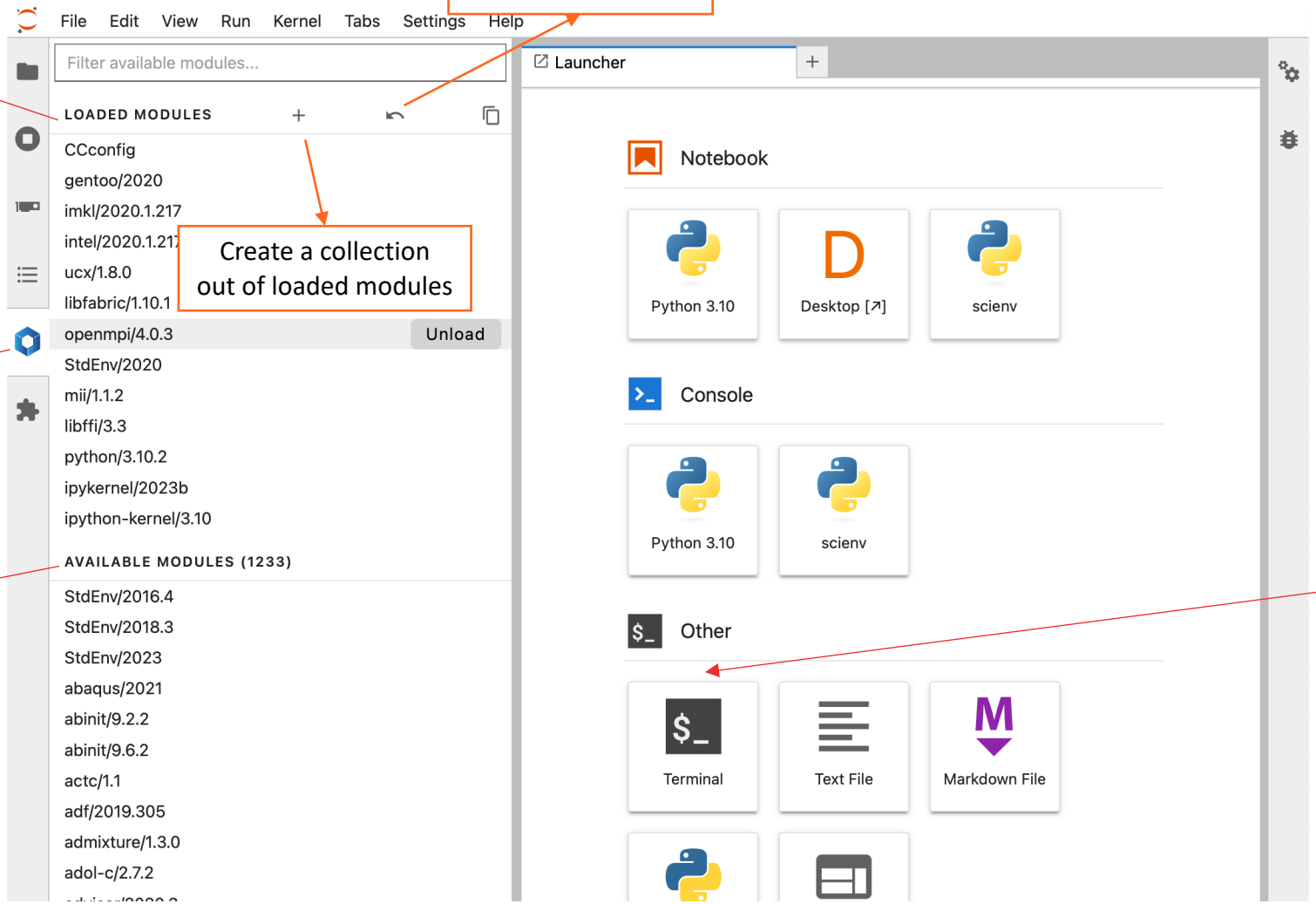
Having commonly used software readily available saves users time and effort

UNIVERSITY OF CALGARY

Digital Infrastructure

Restore a previously saved collection

Loaded modules



Create a collection out of loaded modules

List of HPC modules (LMOD Extention)

Unloaded modules

A saved list of modules is called a “collection”

Using JupyterLab interface
For creating collections could be time consuming.

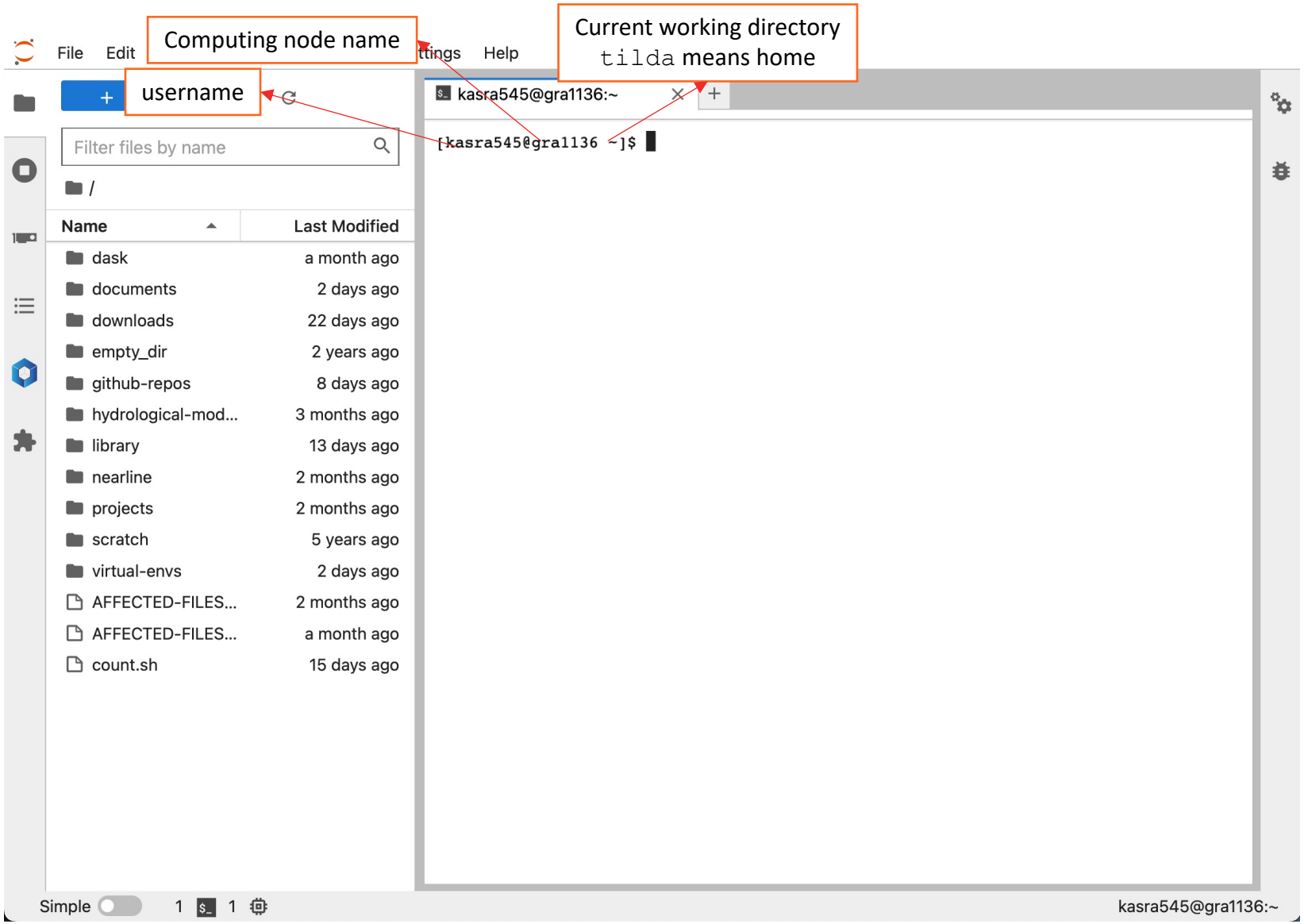
We will use the Terminal to build a collection later.

UNIVERSITY OF CALGARY

Digital Infrastructure

All instructions are available on this presentation's repository:

https://github.com/kasra-keshavarz/community-modelling-workflow-training



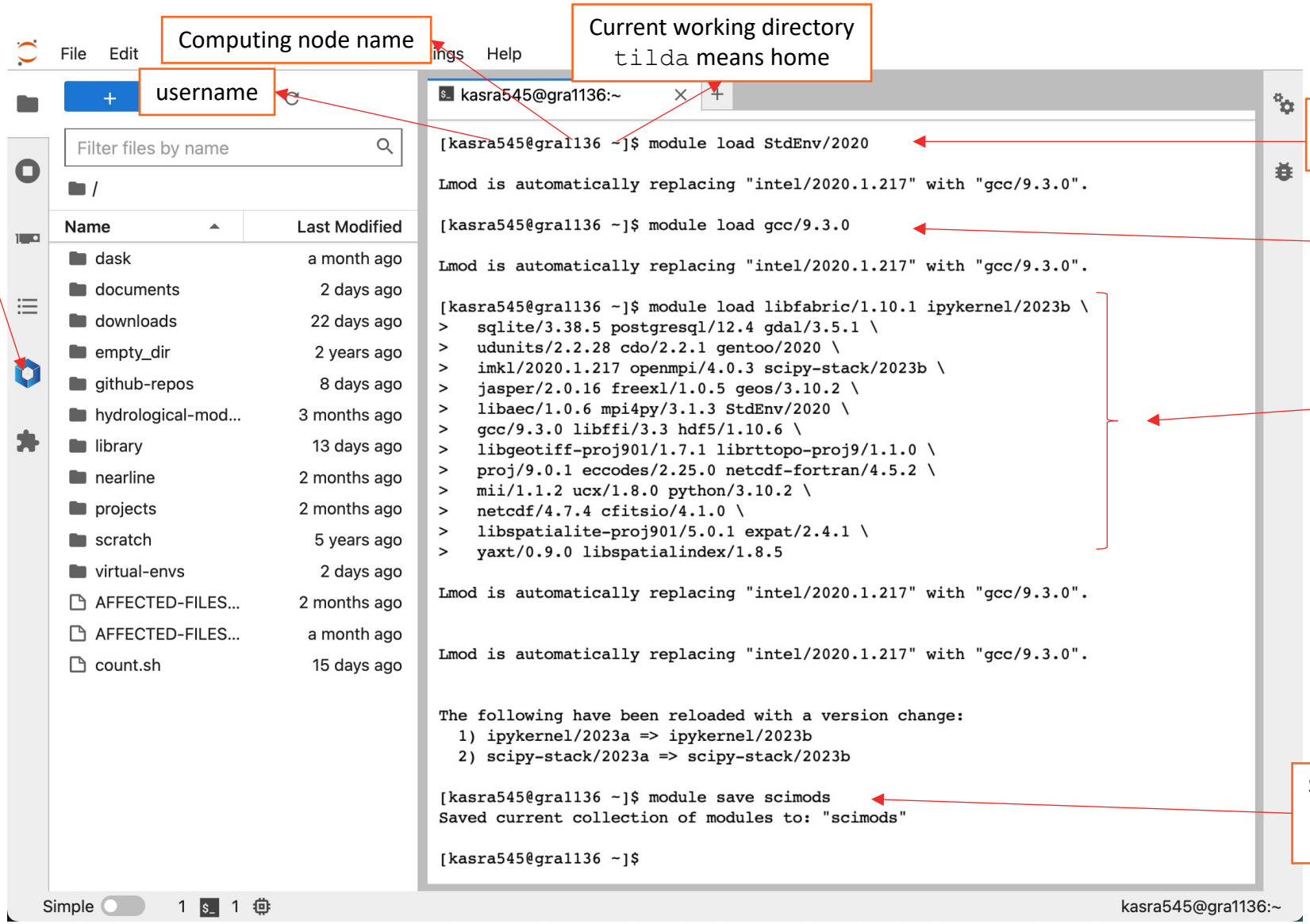
UNIVERSITY OF CALGARY

Digital Infrastructure

Exactly similar to what we can do in this tab

All instructions are available on this presentation's repository:

https://github.com/kasra-keshavarz/community-modelling-workflow-training



Loading "Standard Environment"

Loading Gnu C Compiler version 9.3.0

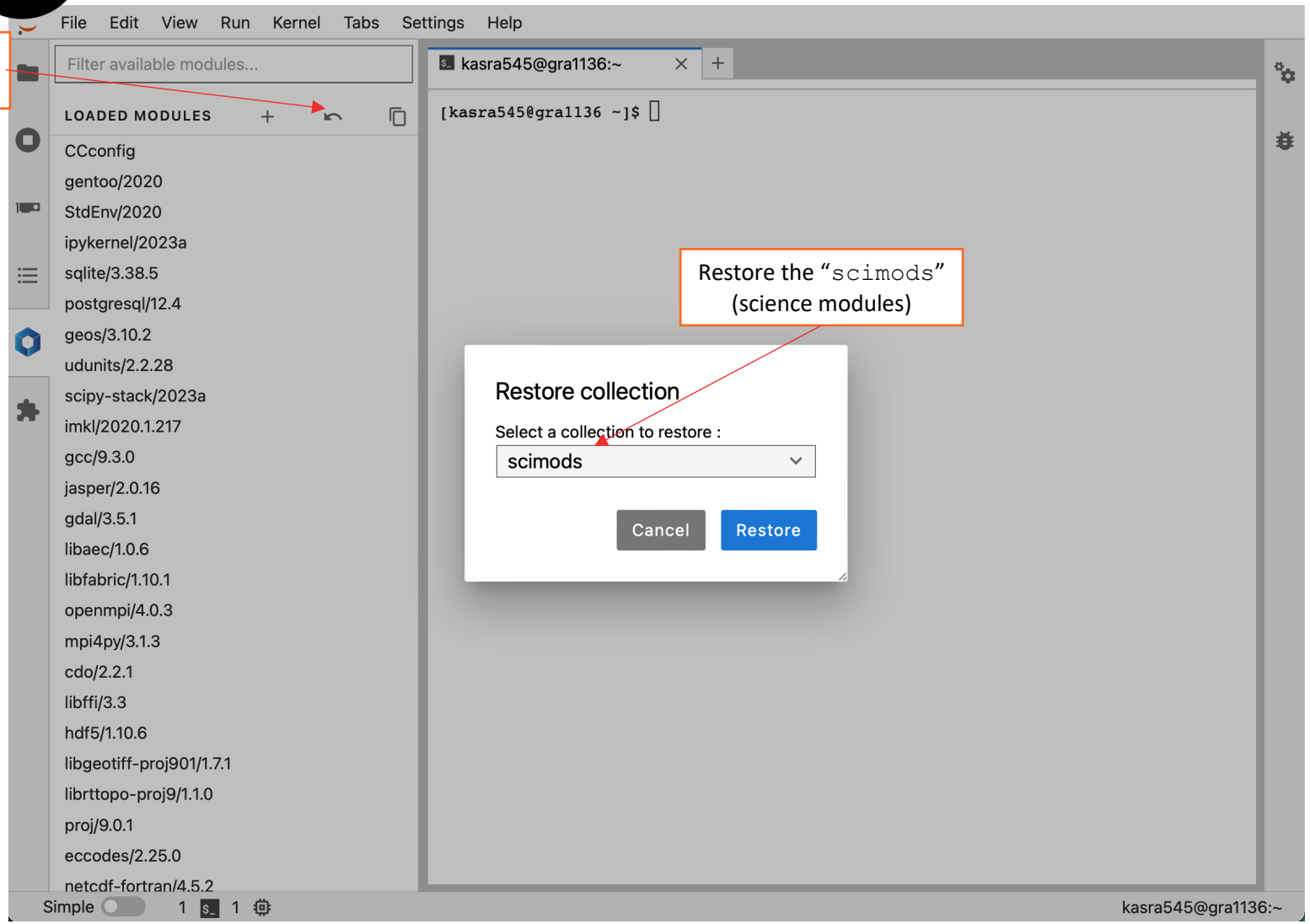
Loading every module we need

Saving all modules as a "collection" called "science modules"

Digital Infrastructure

Refresh the JupyterLab session webpage, and see if you can find the collection you just saved:

Check out what module collections you have



Restore the "`scimods`" (science modules)

Wait for a few seconds so Jupyter loads all modules saved in "`scimods`" collection

This is only a one-time process. You can always access all the collections you saved from Jupyter's LMOD Extention tab

UNIVERSITY OF CALGARY

Any questions so far?

Digital Infrastructure

Python Programming Language

- Community Hydrological Modelling Workflows: much of our workflows are offered in Python package formats
- Python is Powerful: Python always can offer you a useful workflow
- Reproducible Environments: Python's Virtual Environments enables this requirement
- Supported on HPCs and Commercial Cloud regardless of your Operating System: it can be used everywhere
- Graham is no exception: Python is fully supported on Graham

Digital Infrastructure

Before setting up the Python Environment, we need to download the training session's GitHub repository

Digital Infrastructure

High Performance Computing (HPC) systems are comprised of various “nodes”

The idea is to expand horizontally, instead of vertically.

HPC “nodes” are located inside a “rack enclosure”

DALLE generated image

jupyterhub

JupyterHub server lands your JupyterLab session on one these nodes
or, more clearly, “computing” nodes

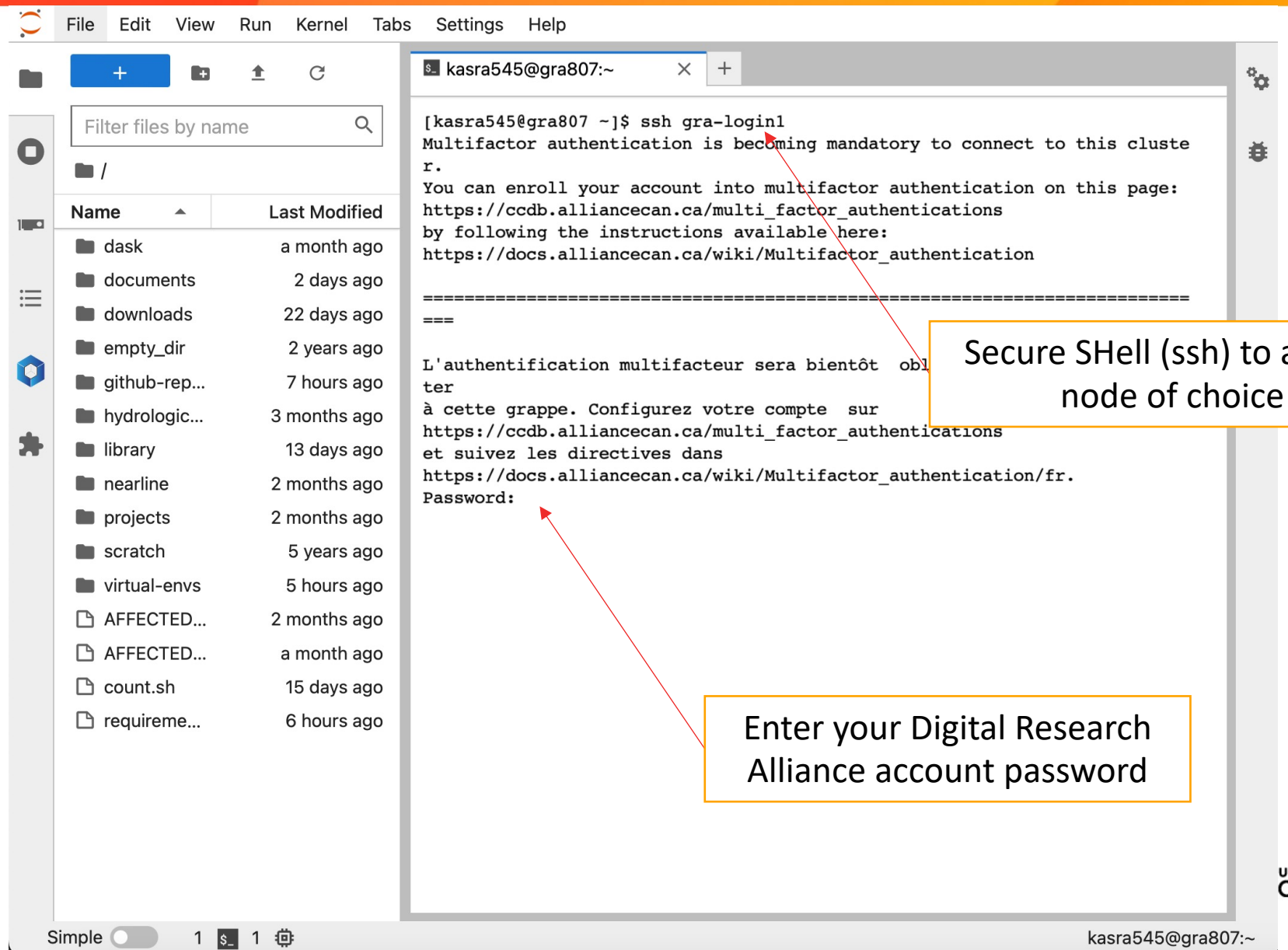
On Digital Research Alliance HPCs, computing nodes do not have access to an active internet connection unfortunately

UNIVERSITY OF CALGARY

Digital Infrastructure

First, we need to switch to the “login node”

Login nodes do have access to an active internet connection



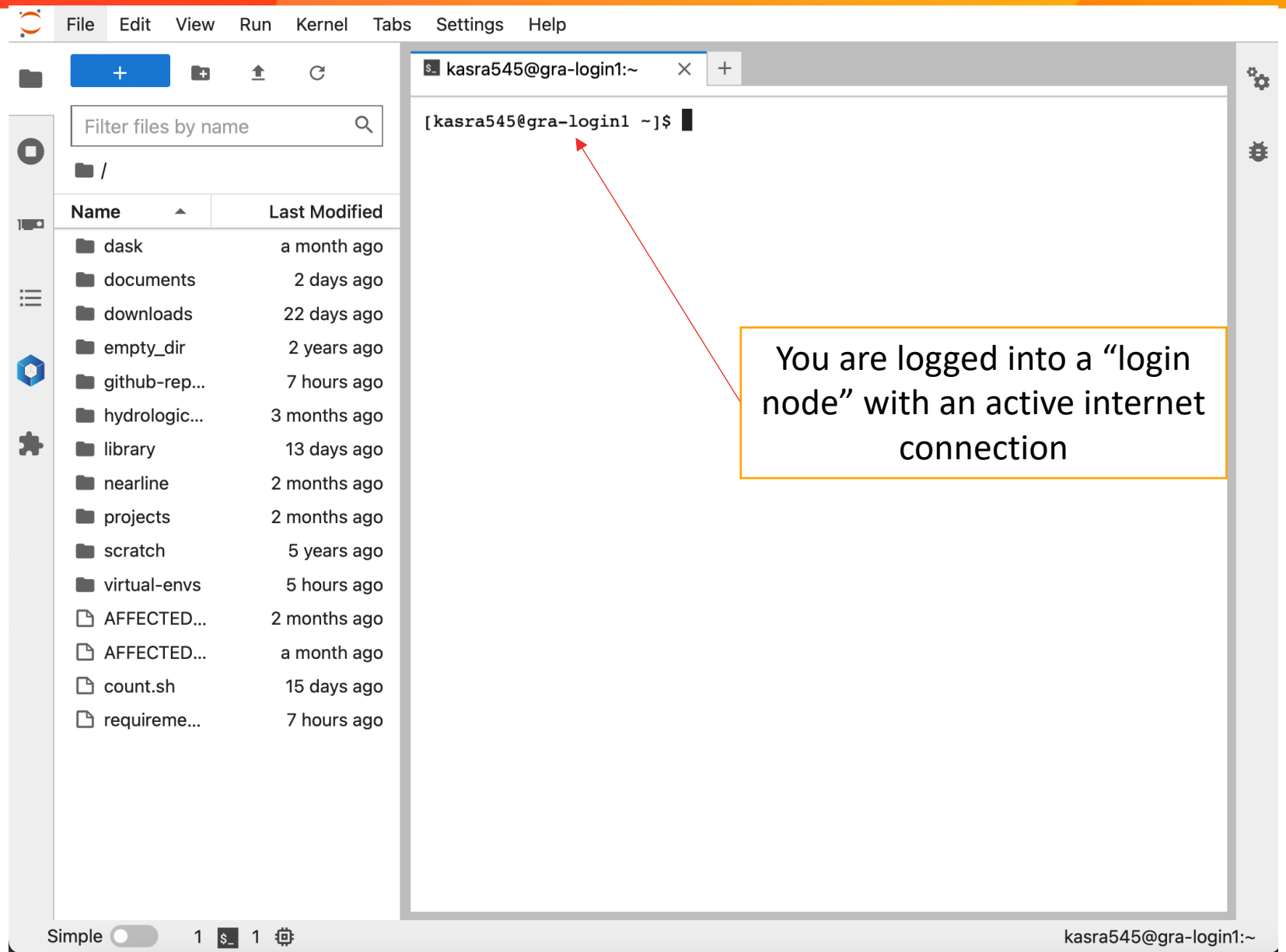
Secure SHell (ssh) to a login node of choice

Enter your Digital Research Alliance account password

UNIVERSITY OF CALGARY

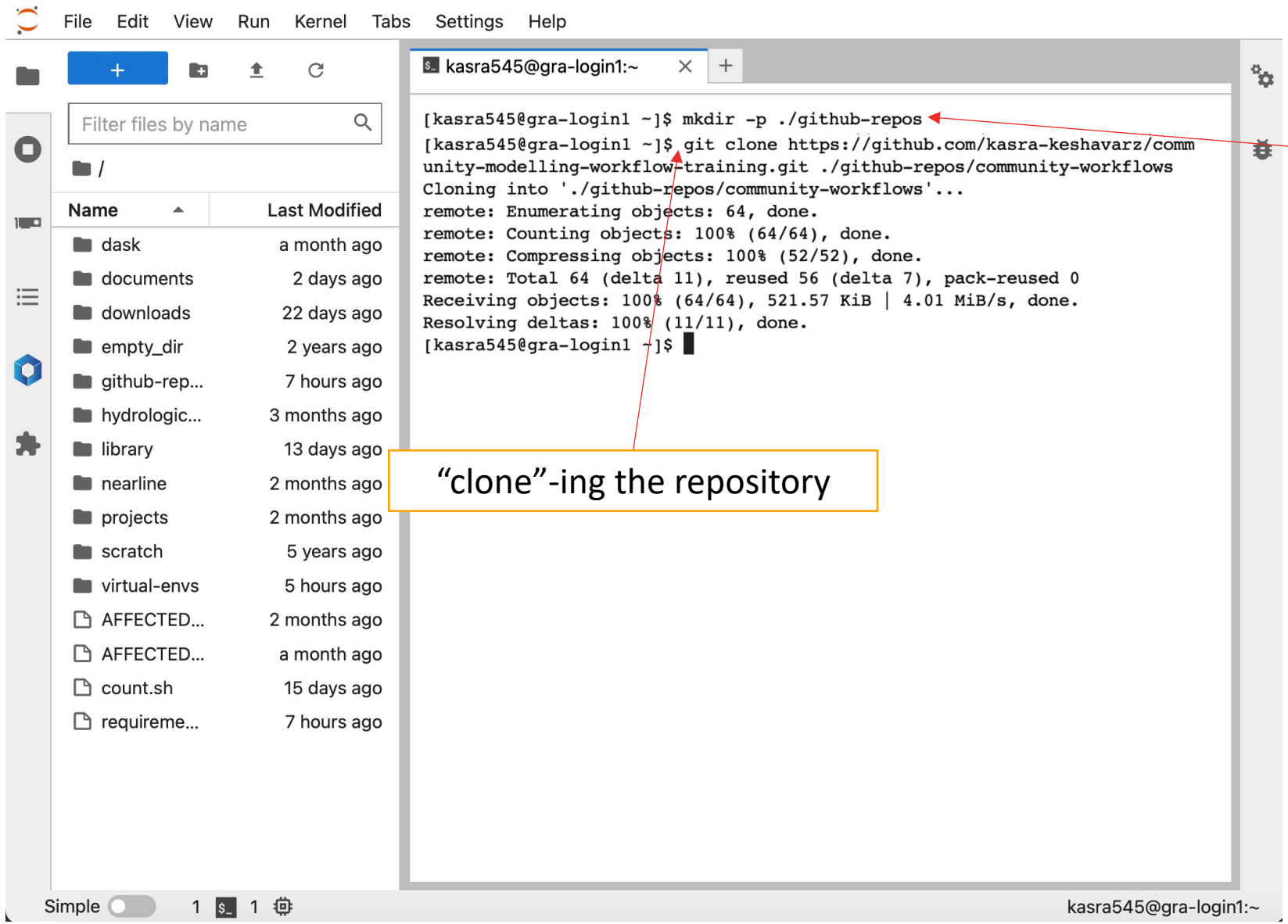
Digital Infrastructure

We use this terminal session now to download the relevant GitHub repository



UNIVERSITY OF CALGARY

Digital Infrastructure



Create a “github-repos” directory to organize downloaded repositories

“clone”-ing the repository

UNIVERSITY OF CALGARY

Digital Infrastructure

1	Only for personal use	Home space 50 GB per user
2	Only for data processing	Scratch space 20 TB per user
3	NOT for personal use – only for shared resources	Project space rrp-kshook: 2.5 PB rrg-mclark: 0.2 PB

DALLE generated image

Any questions so far?

Digital Infrastructure

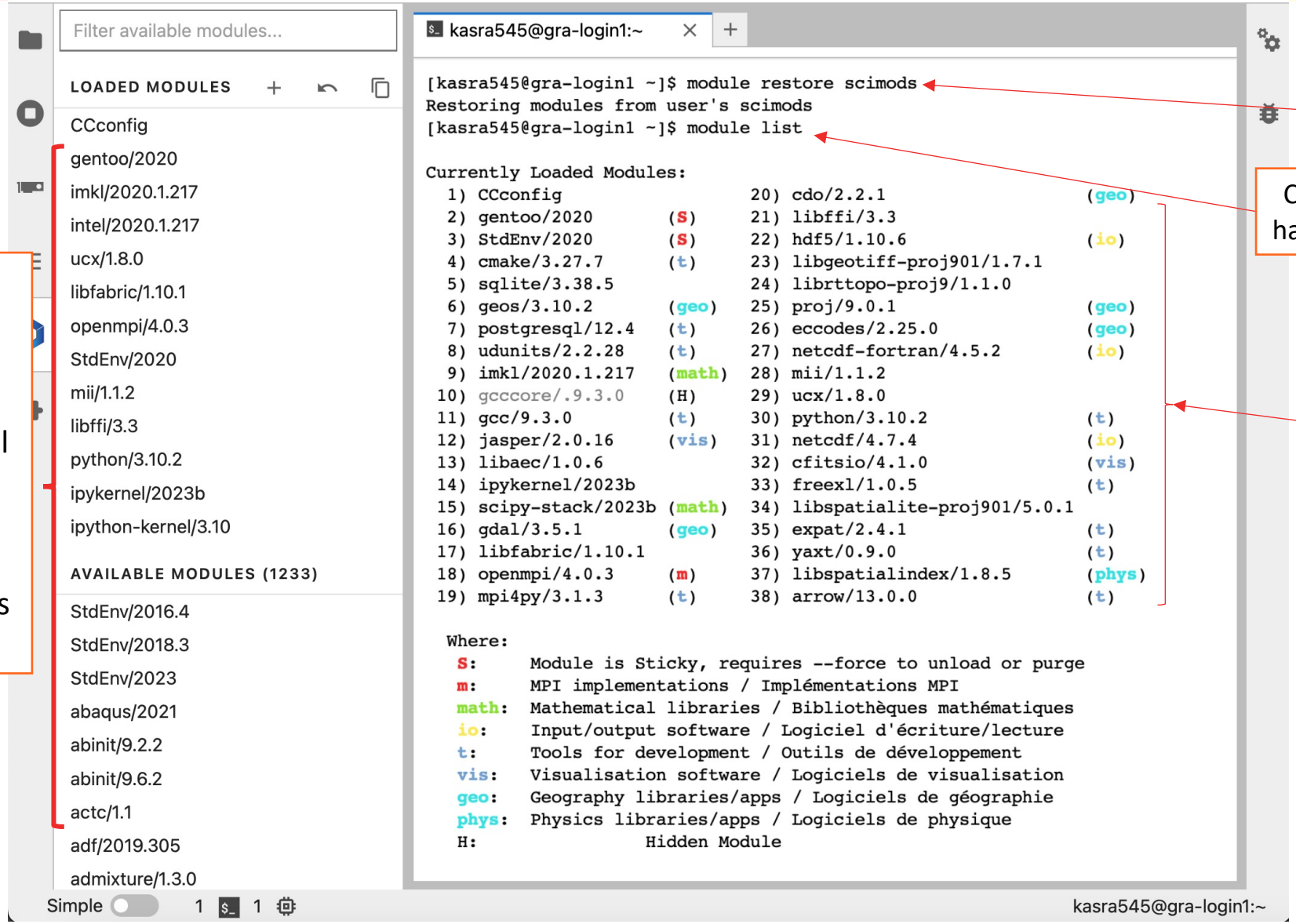
Let’s set up our Python virtual environment

DALLE generated image

UNIVERSITY OF CALGARY

Digital Infrastructure

Although the modules are loaded in your JupyterLab session, Terminal is a separate entity, and we must reload the modules there as well:



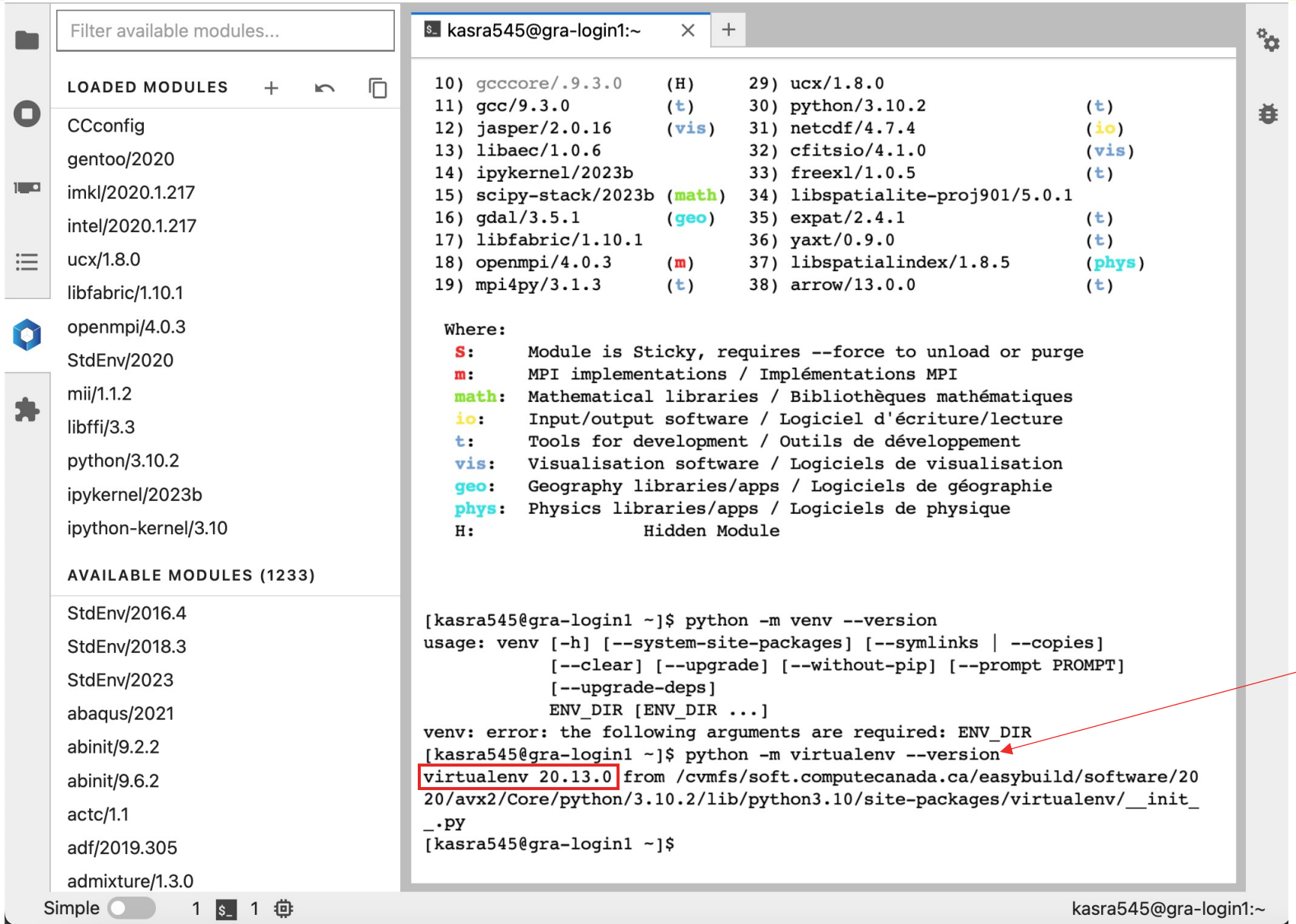
restoring "scimods"

Checking what has been loaded

Is it equal to the list on left?

UNIVERSITY OF CALGARY

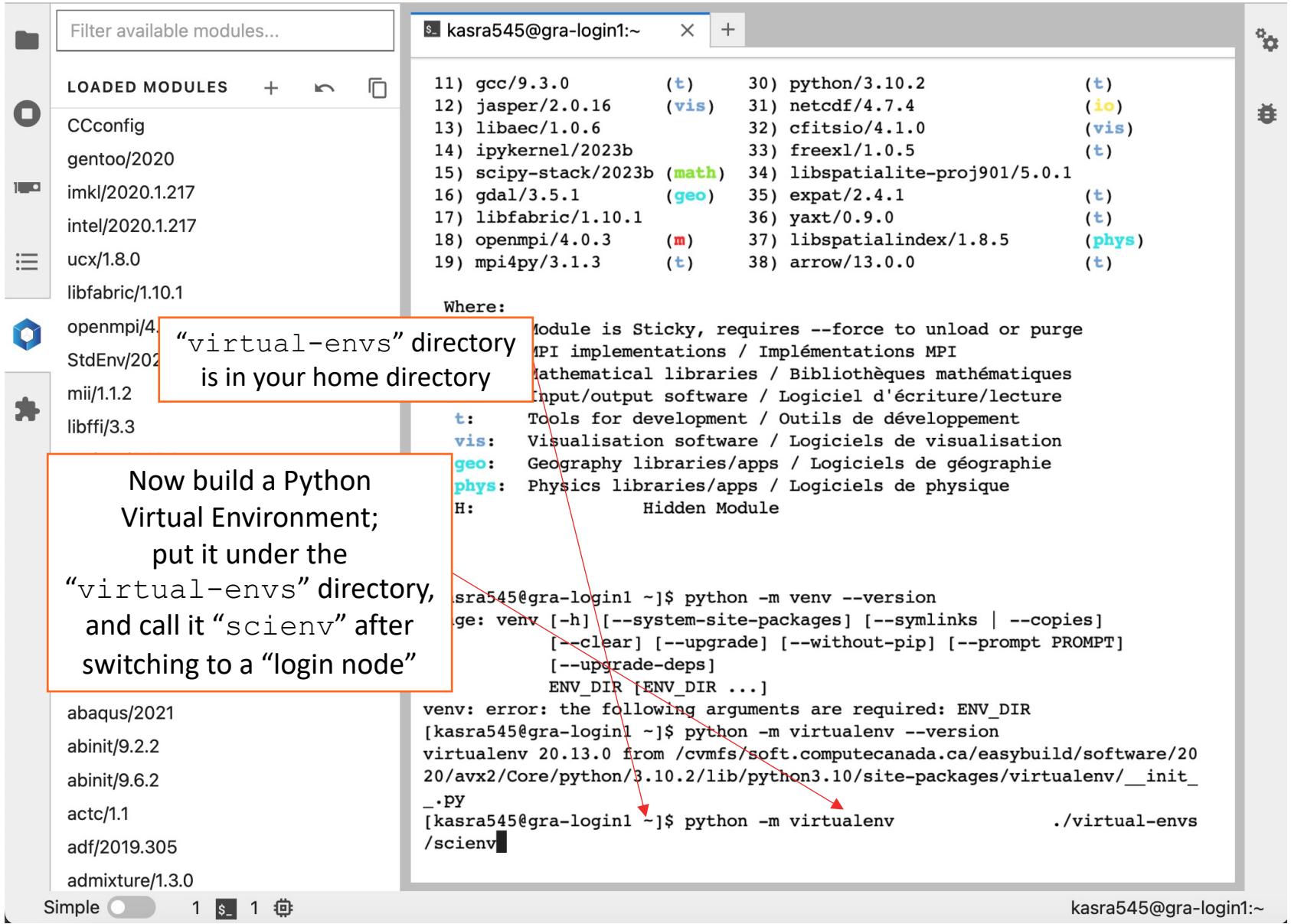
Digital Infrastructure



Are we all using the same version?

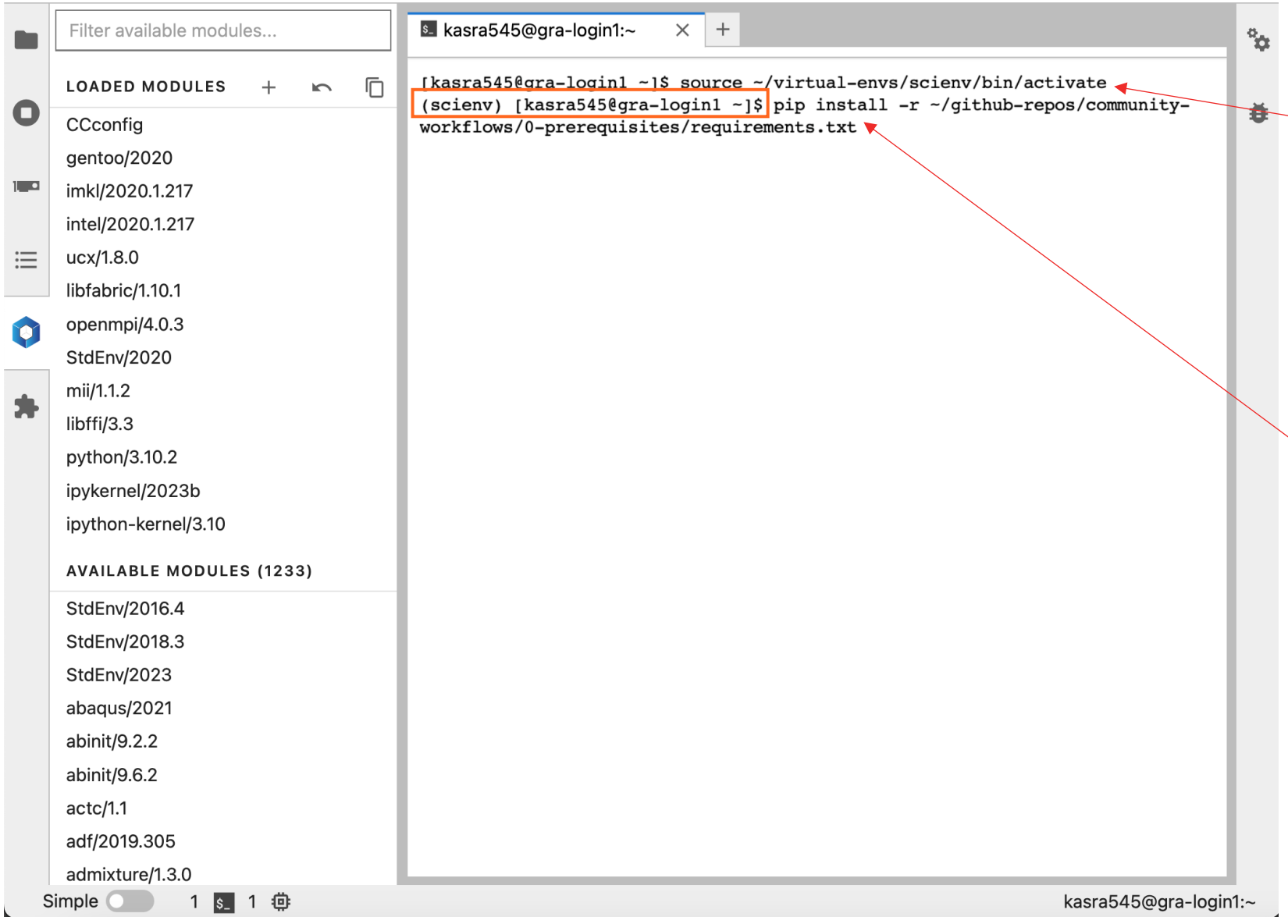
UNIVERSITY OF CALGARY

Digital Infrastructure



UNIVERSITY OF CALGARY

Digital Infrastructure

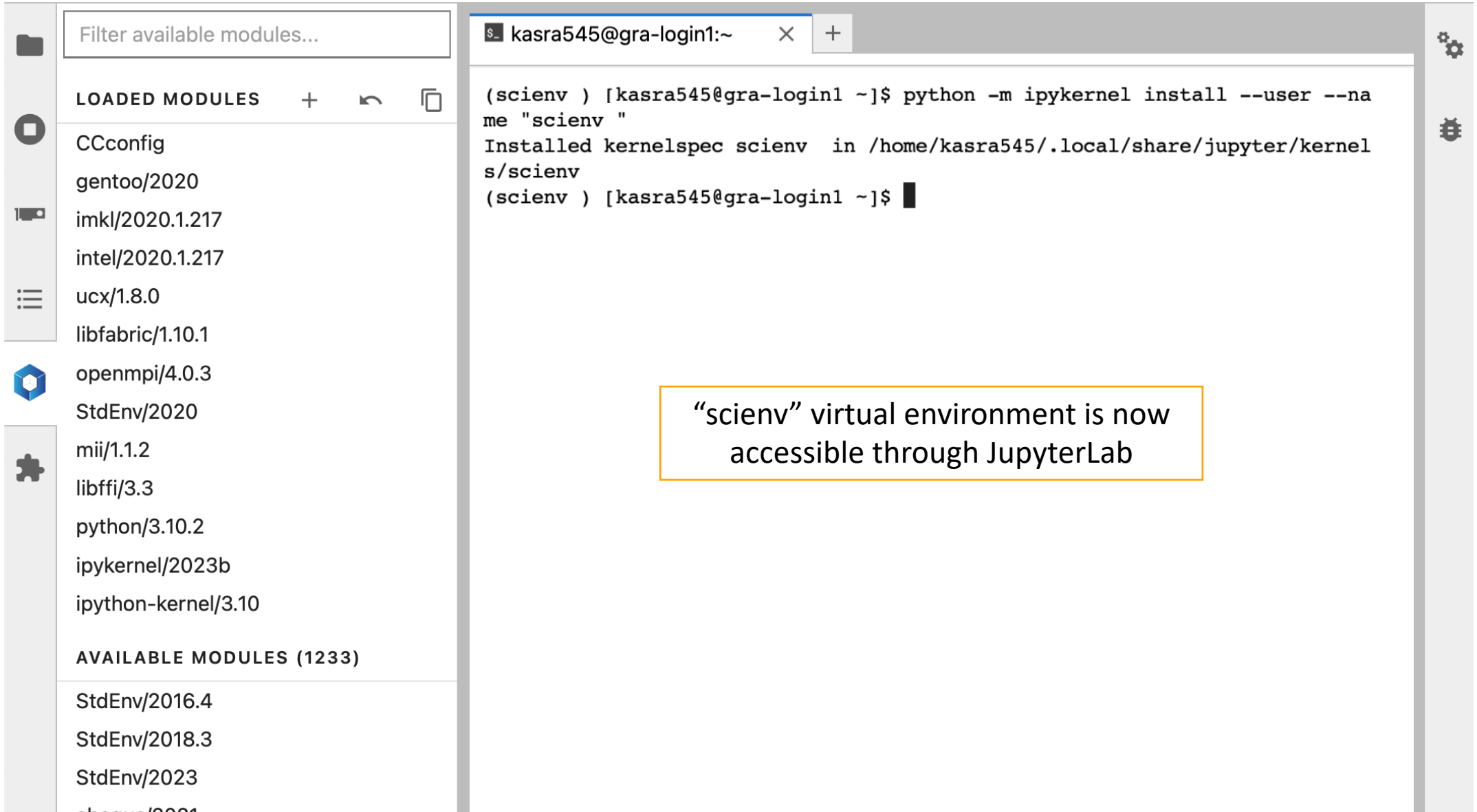


After creation, let's activate the virtual environment

Install the packages included in the training session's repository

UNIVERSITY OF CALGARY

Digital Infrastructure

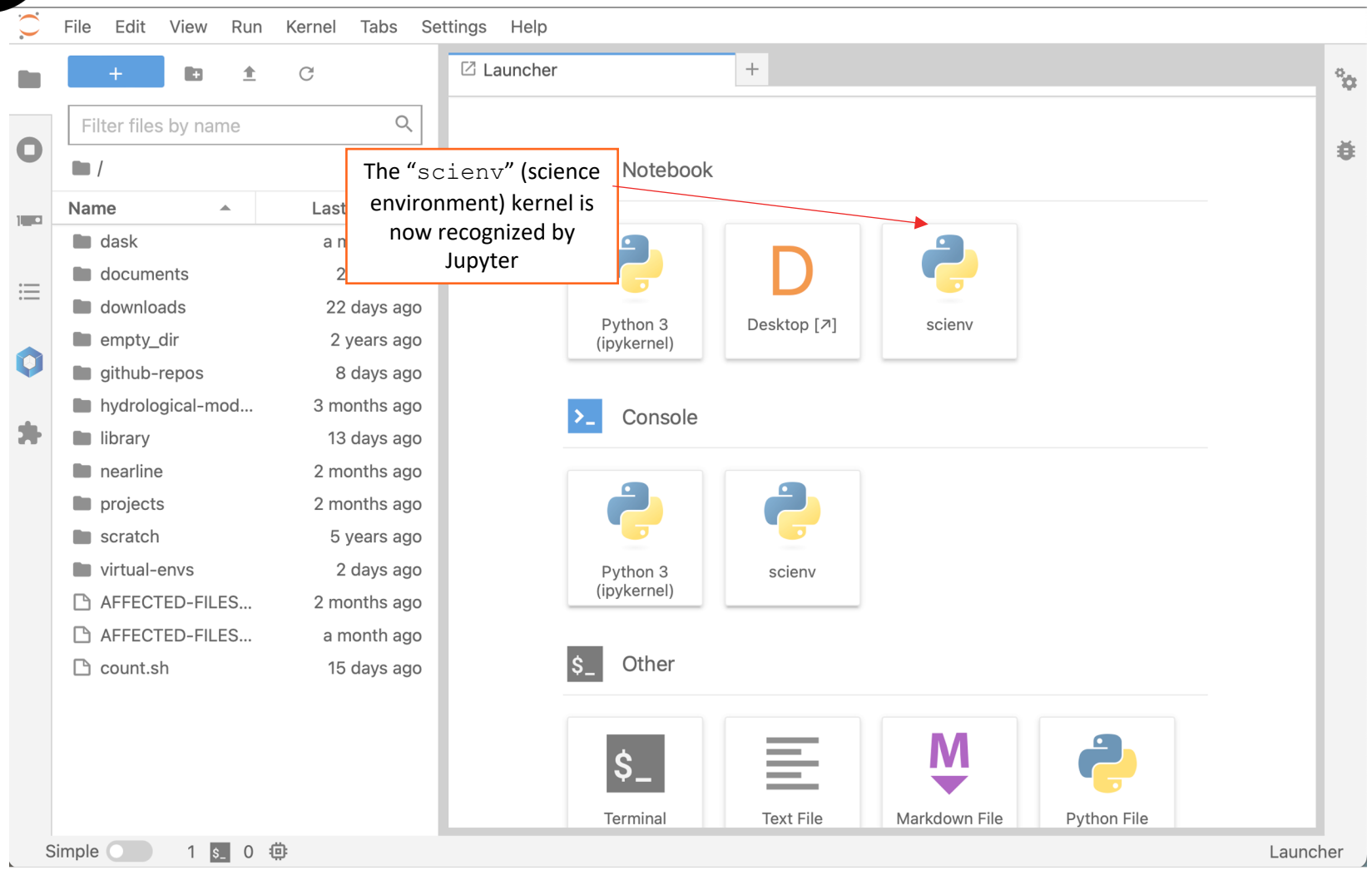


"scienv" virtual environment is now accessible through JupyterLab

UNIVERSITY OF CALGARY

Digital Infrastructure

Refresh the JupyterLab session webpage, and see if you can find the collection you just saved:

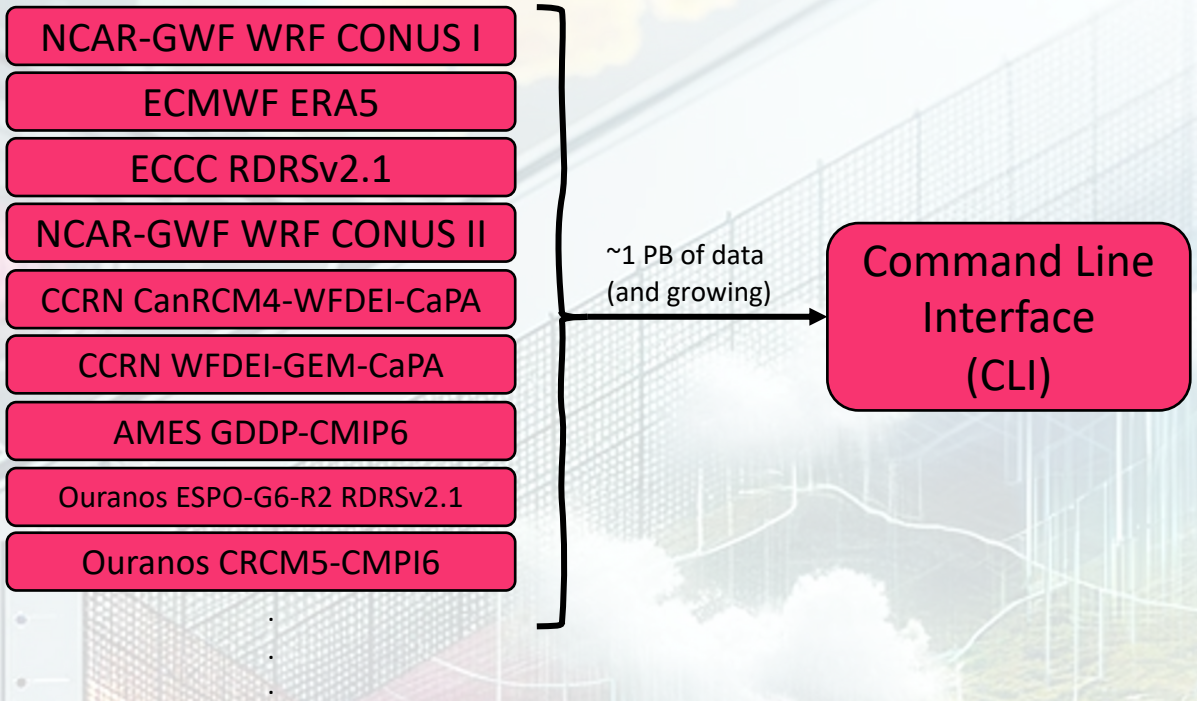


UNIVERSITY OF CALGARY

Outline

- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Introducing Community Workflows & Tools




```
./extract-dataset.sh  --dataset=CONUS1 \
  --dataset-dir="/project/rpp-kshook/Model_Output/WRF/CONUS/CTRL" \
  --output-dir="$HOME/scratch/conus_i_output/" \
  --start-date="2000-10-01 00:00:00" \
  --end-date="2013-09-30 23:00:00" \
  --lat-lims=49,54  \
  --lon-lims=-120,-98 \
  --variable="T2,PREC_ACC_NC,Q2,ACSWDNB,ACLWDNB,U10,V10,PSFC" \
  --prefix="conus_i" \
  --email="youremail@company.ca" \
  -j;
```

Steps:

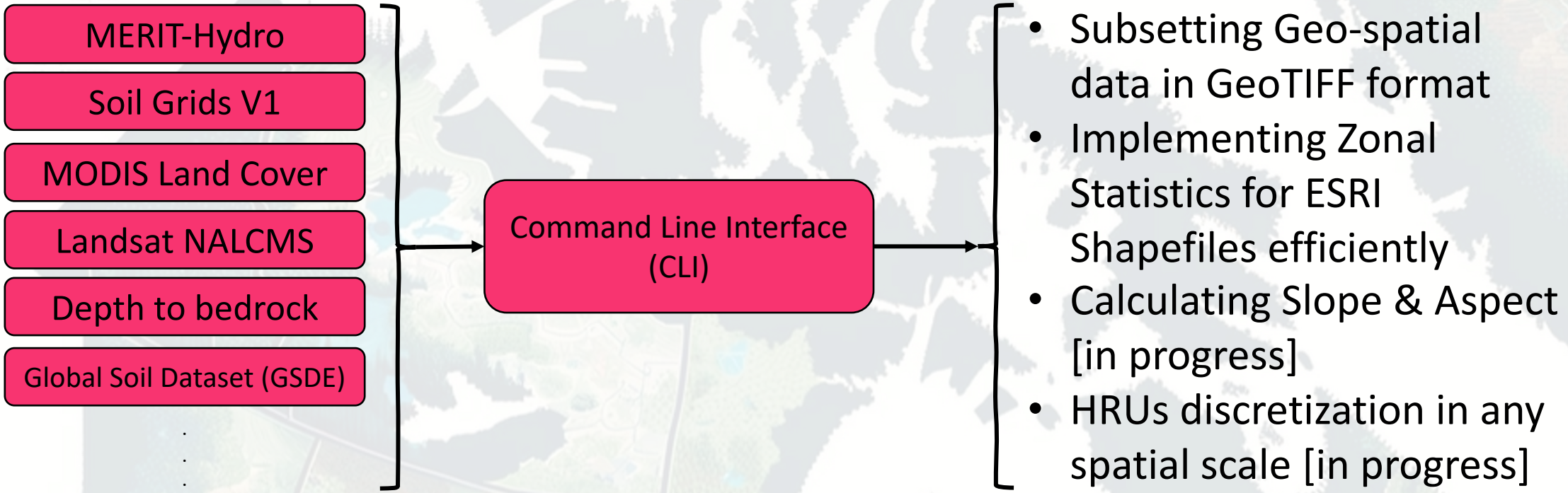
1. Login to Digital Alliance of Canada (formerly Compute Canada) Graham cluster,
2. Make sure you have access to the designated project space and allocation,
3. Clone the remote GitHub repository,
4. Subset any spatial and temporal extents of interest for any of the available datasets with only **one line of code**.

https://github.com/kasra-keshavarz/datatool

UNIVERSITY OF CALGARY

DALL-E generate image

Introducing Community Workflows & Tools



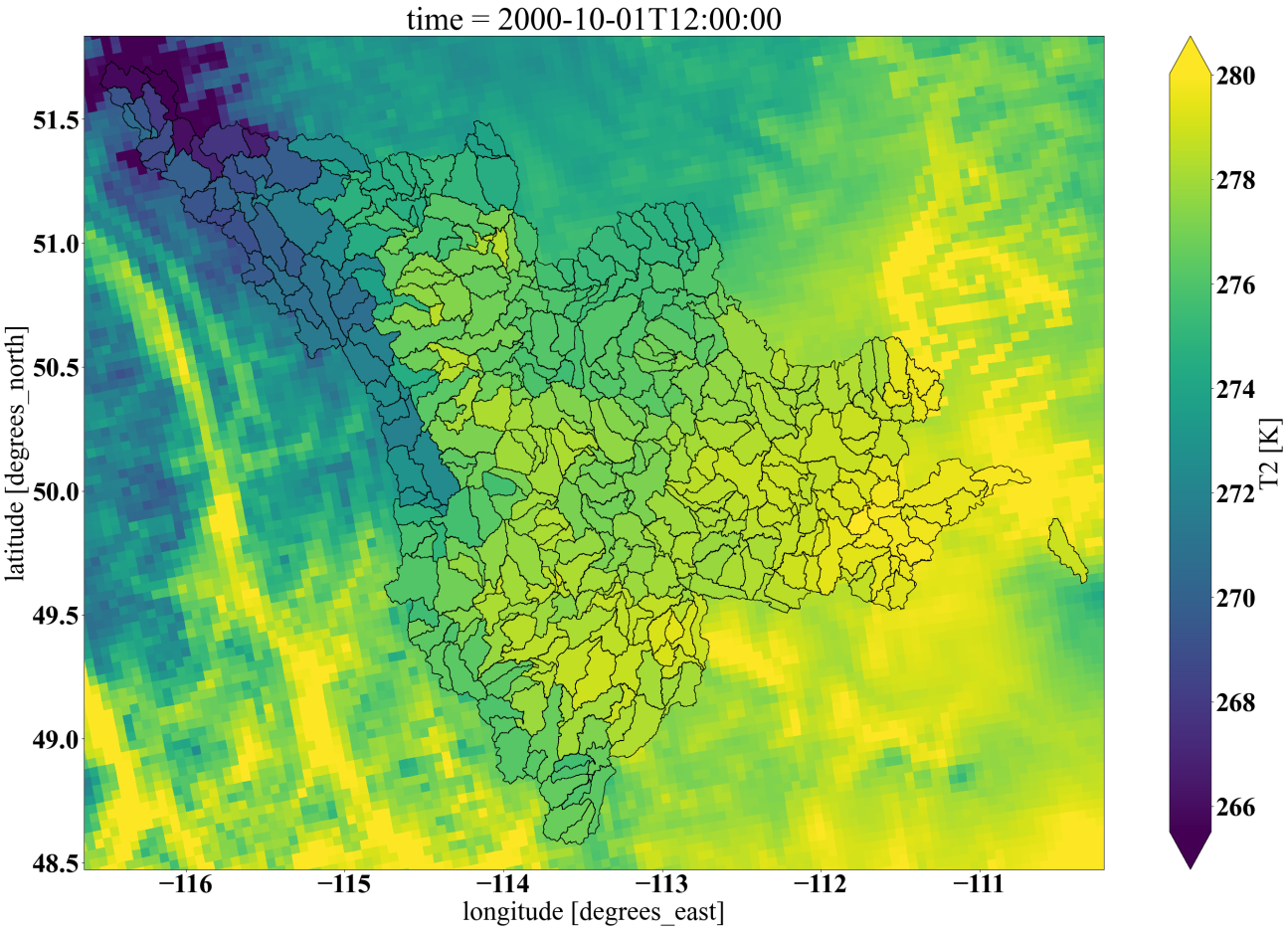
Steps:

1. Login to Digital Alliance of Canada (formerly Compute Canada) Graham cluster,
2. Make sure you have access to the designated project space and allocation,
3. Clone the repository,
4. Subset and/or implement zonal statistics for any spatial and temporal extents of interest for any of the included datasets with only **one line of code**.

https://github.com/kasra-keshavarz/gistool

UNIVERSITY OF CALGARY

Introducing Community Workflows & Tools




```
kasras-mbp:easymore ShervanGharari$ easymore
Usage: easymore [OPTIONS] COMMAND [ARGS]...

  EASYMORE is a collection of functions that allows extraction of the
  data from a NetCDF file for a given shapefile such as a basin,
  catchment, points or lines. It can map gridded data or model output to
  any given shapefile and provide area average for a target variable.

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  cli   Run Easymore using CLI
  conf  Run Easymore using a JSON configuration file

  For bug reports, questions, and discussions open an issue at
  https://github.com/ShervanGharari/EASYMORE.git
```

Command Line Interface (CLI)
and
Python Library

https://github.com/ShervanGharari/EASYMORE.git

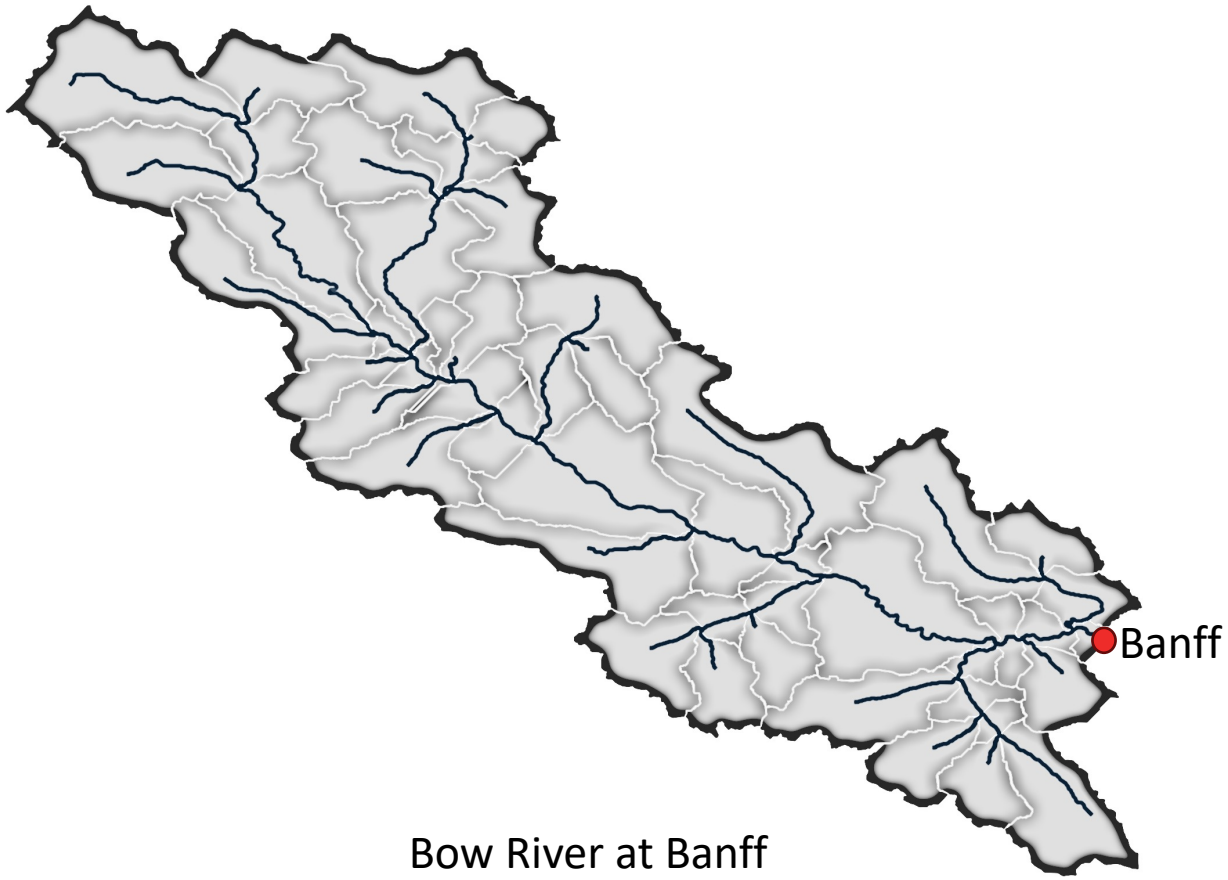
UNIVERSITY OF CALGARY

Any questions so far?

Outline

- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Practice Example – Bow River at Banff



Bow River at Banff

Sub-basins and river network extracted from `MERIT-Basins` using `Hydrant v0.1.0-dev0`

All instructions are available on this presentation's repository:

https://github.com/kasra-keshavarz/community-modelling-workflow-training

UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff

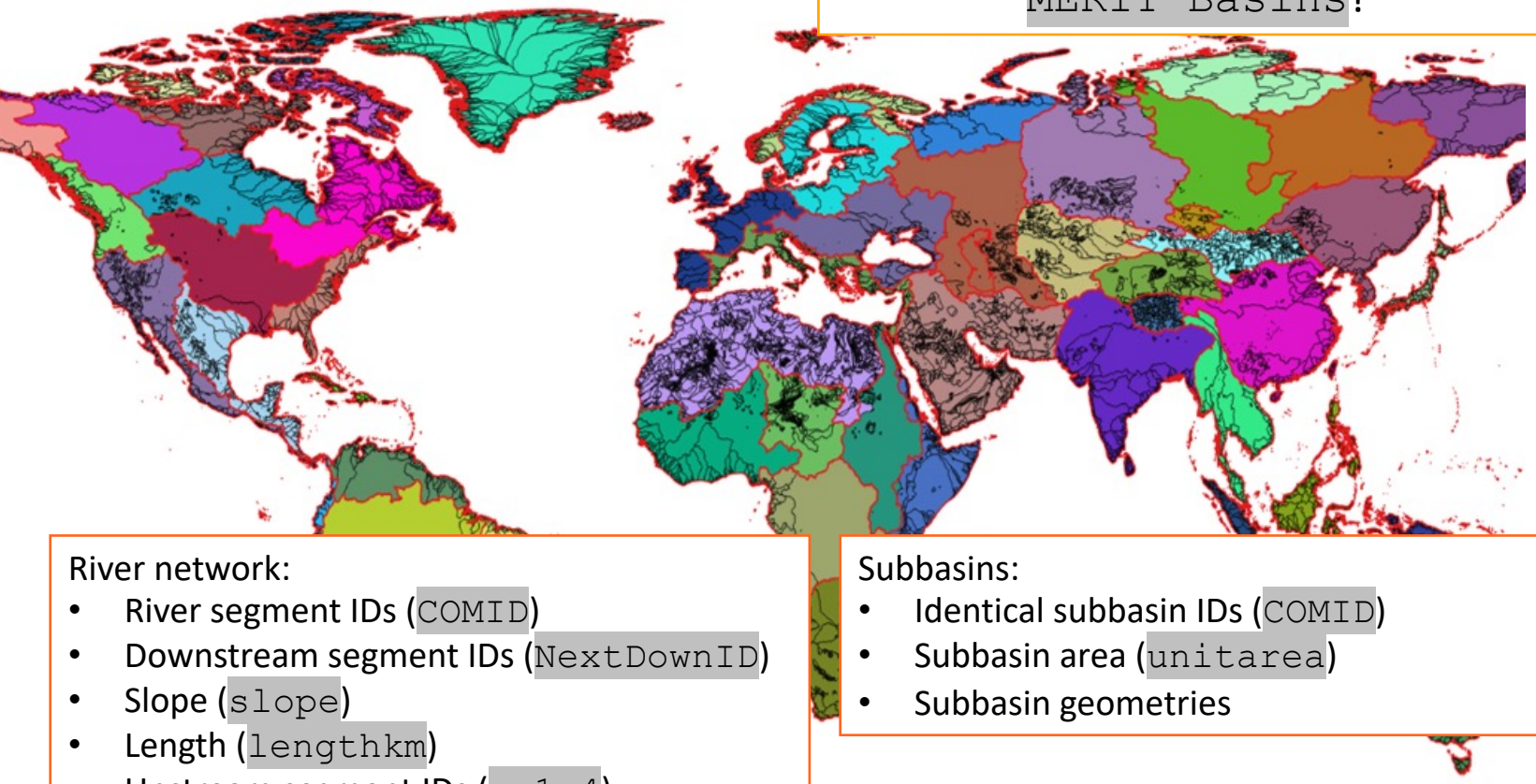
First step is to extract a geospatial fabric for the area of interest

UNIVERSITY OF CALGARY

DALLE generated image

Practice Example – Bow River at Banff

What information is provided through `MERIT-Basins`?



- 25 km^2 drainage area threshold for channels
- Corrected/refined basin/region definitions
- Below 25 km^2 non-channelized areas along the coast or some endorheic areas (incomplete catchments or hillslopes) excluded
- ~2.94 million river reaches (unit catchments)

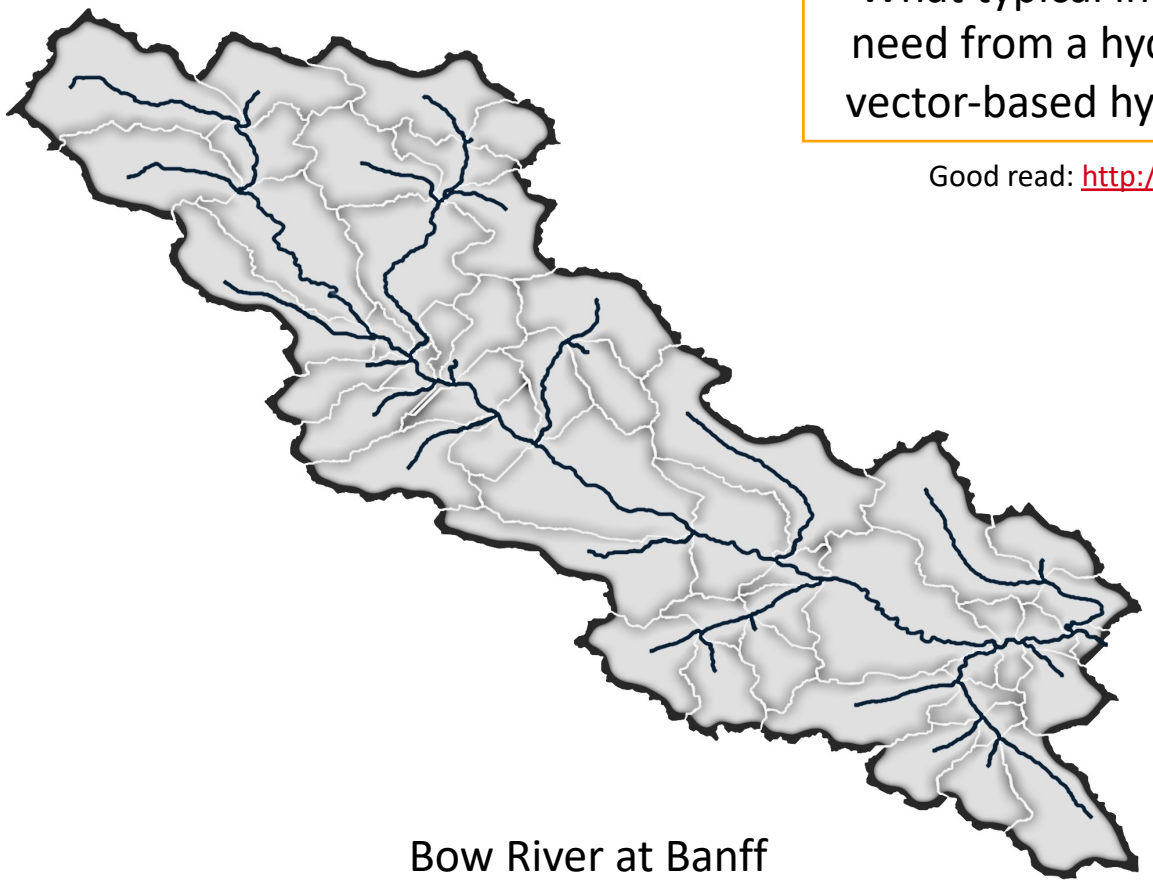
River network:
- River segment IDs (`COMID`)
- Downstream segment IDs (`NextDownID`)
- Slope (`slope`)
- Length (`lengthkm`)
- Upstream segment IDs (`up1-4`)
- Upstream drainage area (`uparea`)
- River geometries

Subbasins:
- Identical subbasin IDs (`COMID`)
- Subbasin area (`unitarea`)
- Subbasin geometries

Non-contributing areas:
- Non-contributing area IDs (`FID`)
- Non-contributing area geometries

`MERIT-Basins` webpage: https://www.reachhydro.org/home/params/merit-basins

Practice Example – Bow River at Banff



Bow River at Banff

Sub-basins and river network extracted from `MERIT-Basins` using `Hydrant v0.1.0-dev0`

What typical information we usually need from a hydrography dataset for vector-based hydrological modelling?

Good read: http://switchfromshapefile.org

River network:

- River segment ID values
- Immediate downstream segment ID values
- Length

Depending on your routing scheme you may need:

- Slope [optional]
- Width [optional]
- etc.

Subbasins:

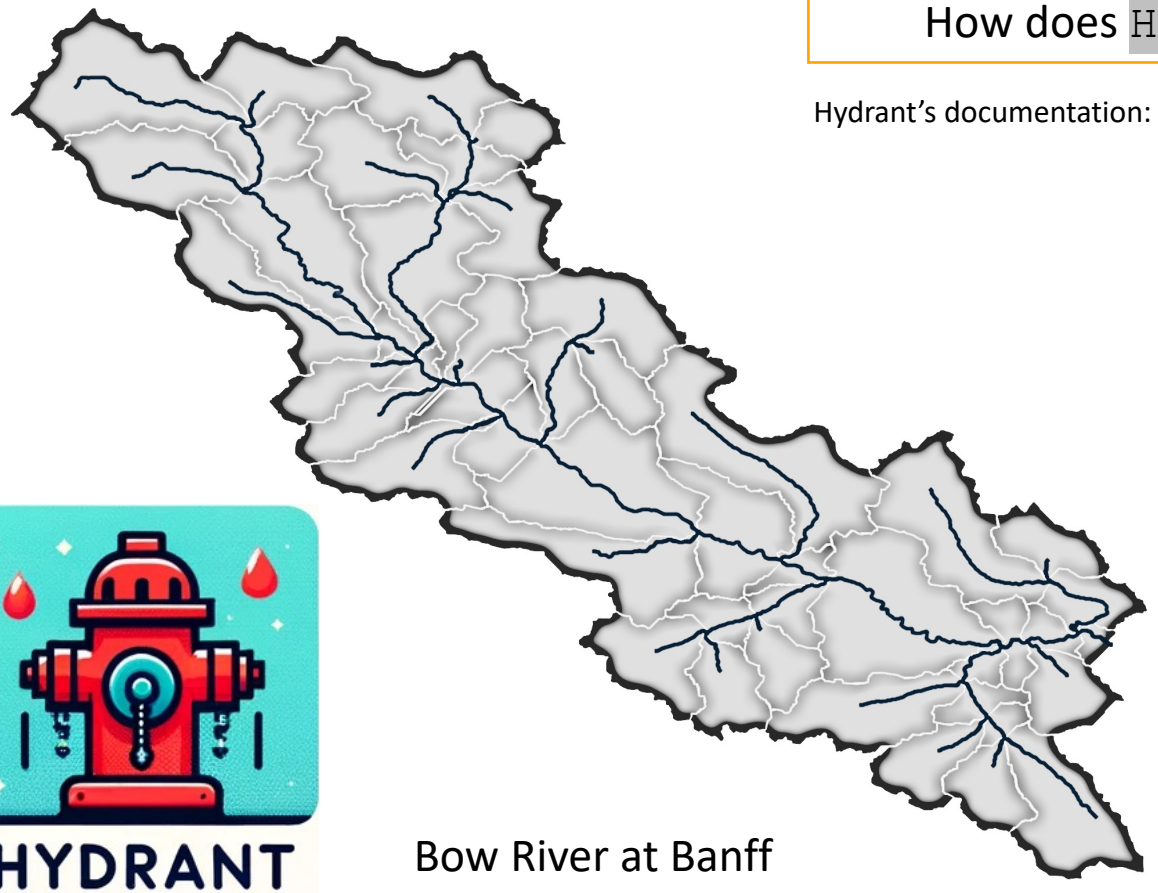
- Subbasin ID values
- Correspondence Mapping Table between Subbasin and River Segment IDs (could be identical value)
- Subbasin Area [optional]
- etc.

Practice Example – Bow River at Banff

How does `HydrAnT` help us?

Hydrant's documentation: https://hydrantpy.readthedocs.org

NetworkX
Network Analysis in Python



Bow River at Banff

Sub-basins and river network extracted from `MERIT-Basins` using `Hydrant v0.1.0-dev0`

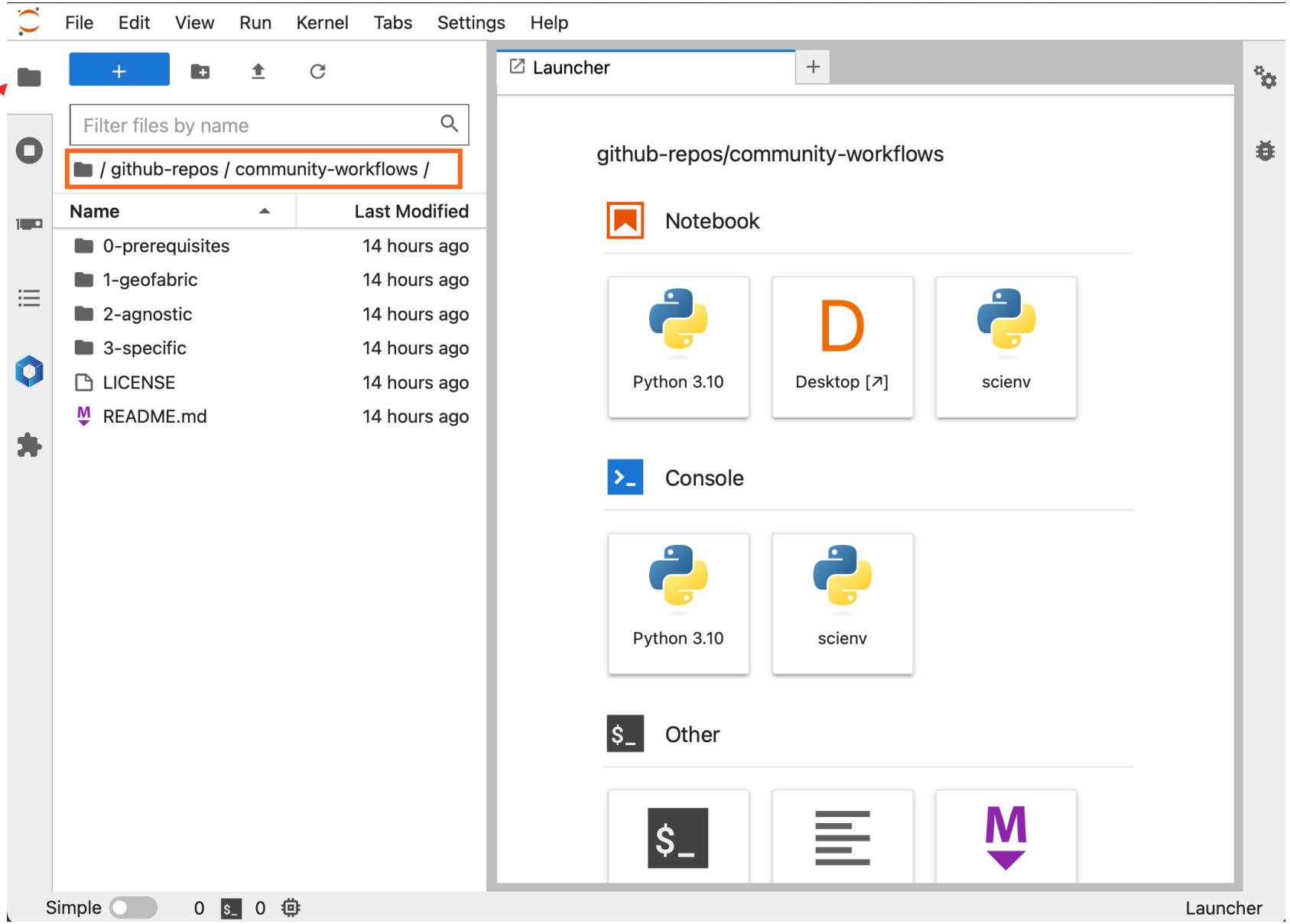
- Extracting subsets of hydrography datasets given a point or a boundary area,
- Hydrant is data-agnostic; if the input hydrography is valid and standard, you can use `Hydrant`,
- Common sanity checks for datasets:
  - Is water going upstream?
  - Is there a cycle of river segments?
  - etc.
- Aggregation methods to dissolve elements of hydrography datasets
- Knowledge extraction:
  - What is the longest branch of river segments?
  - What are the upstream segment of a certain point in the river network?
  - What are the downstream segments of a river segment?
  - What are the Strahler order of river segment?

UNIVERSITY OF CALGARY

Any questions so far?

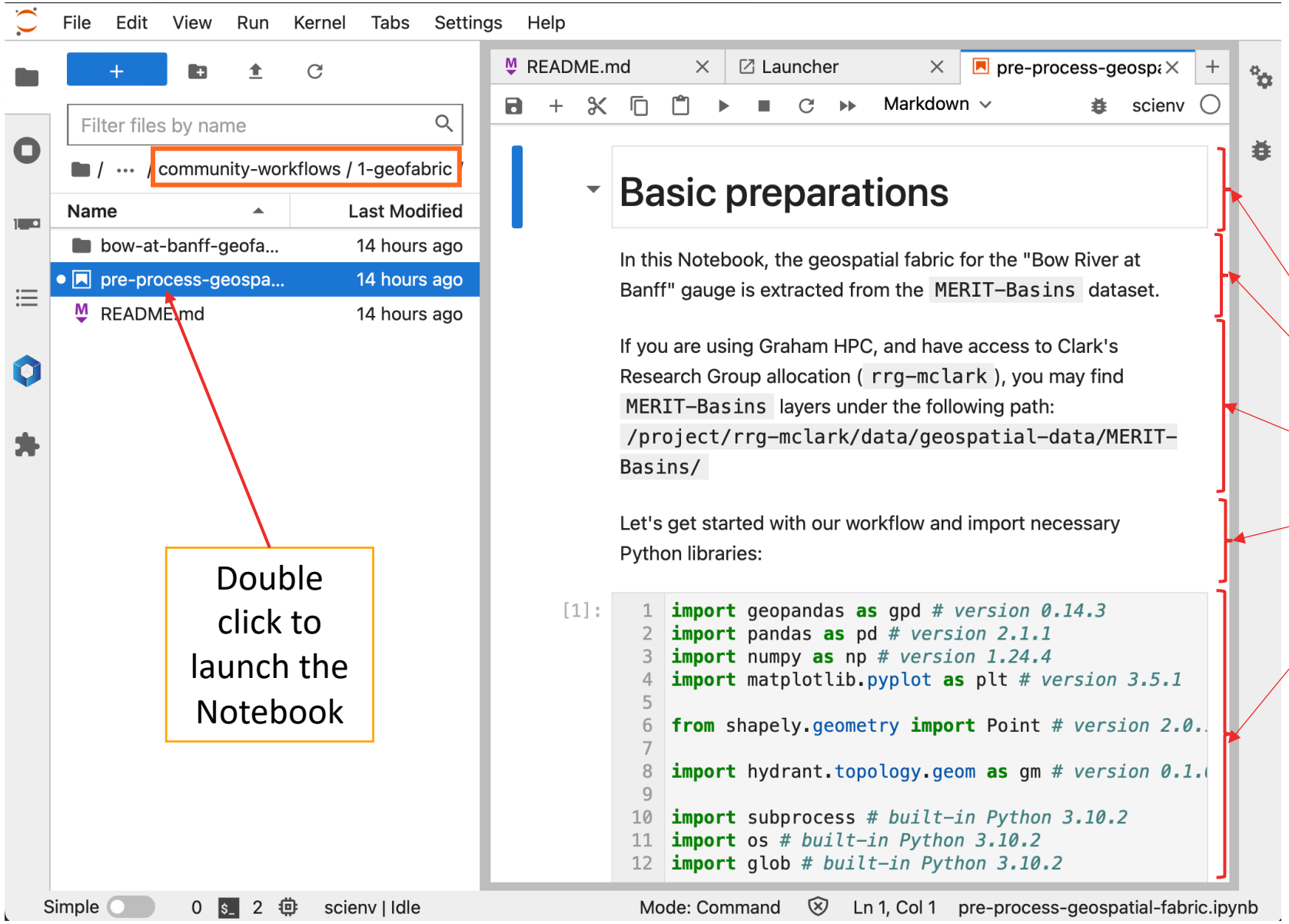
Practice Example – Bow River at Banff

Navigate using JupyterLab file browser



UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff



Double click to launch the Notebook

Use “Shift + Enter” to execute Notebook cells

Notebook cells

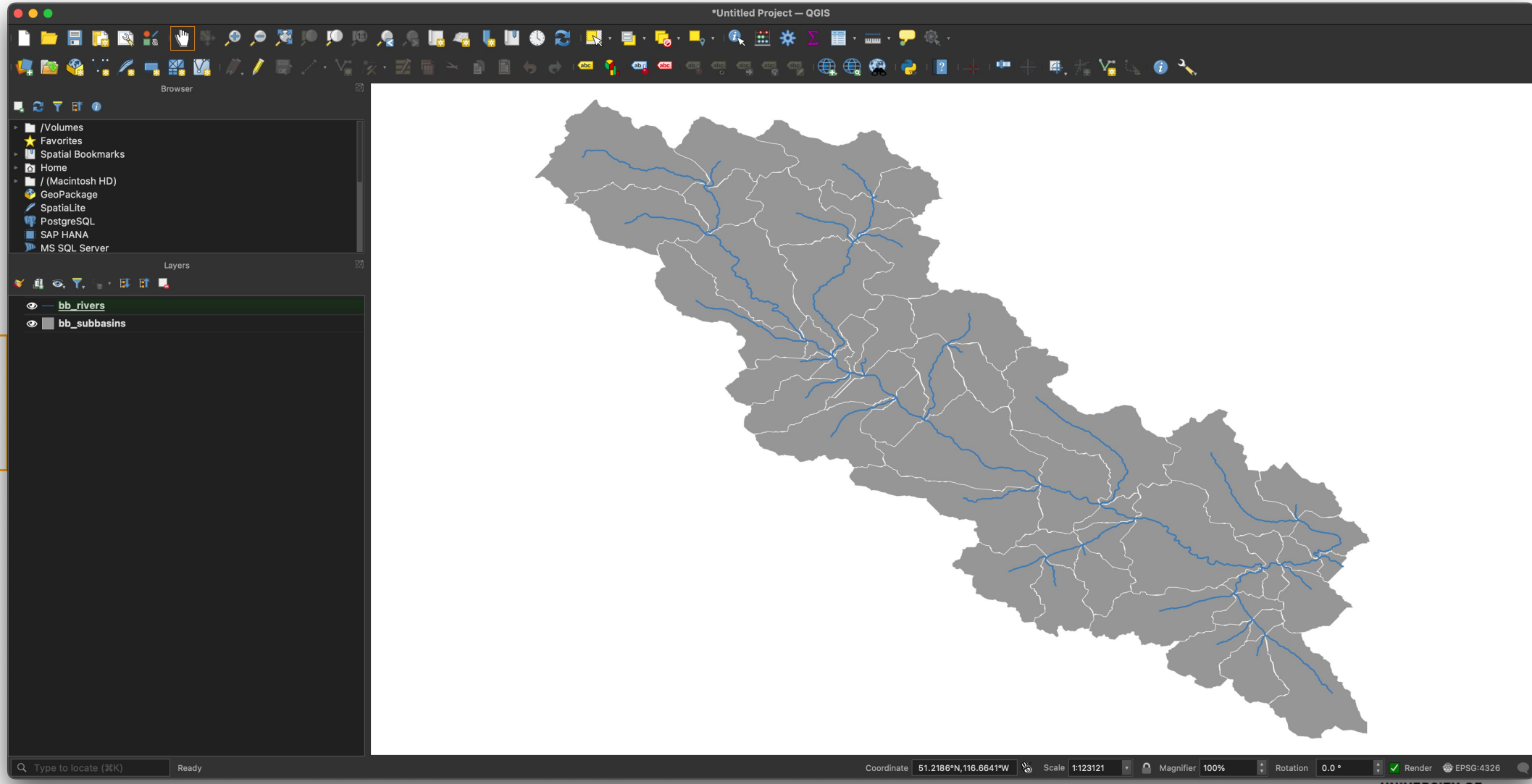
Tip: review keyboard shortcuts!

Practice Example – Bow River at Banff

Switching to the relevant Jupyter Notebook webpage

Practice Example – Bow River at Banff

You can view the extracted geospatial fabric in a typical GIS program, such as *QGIS*



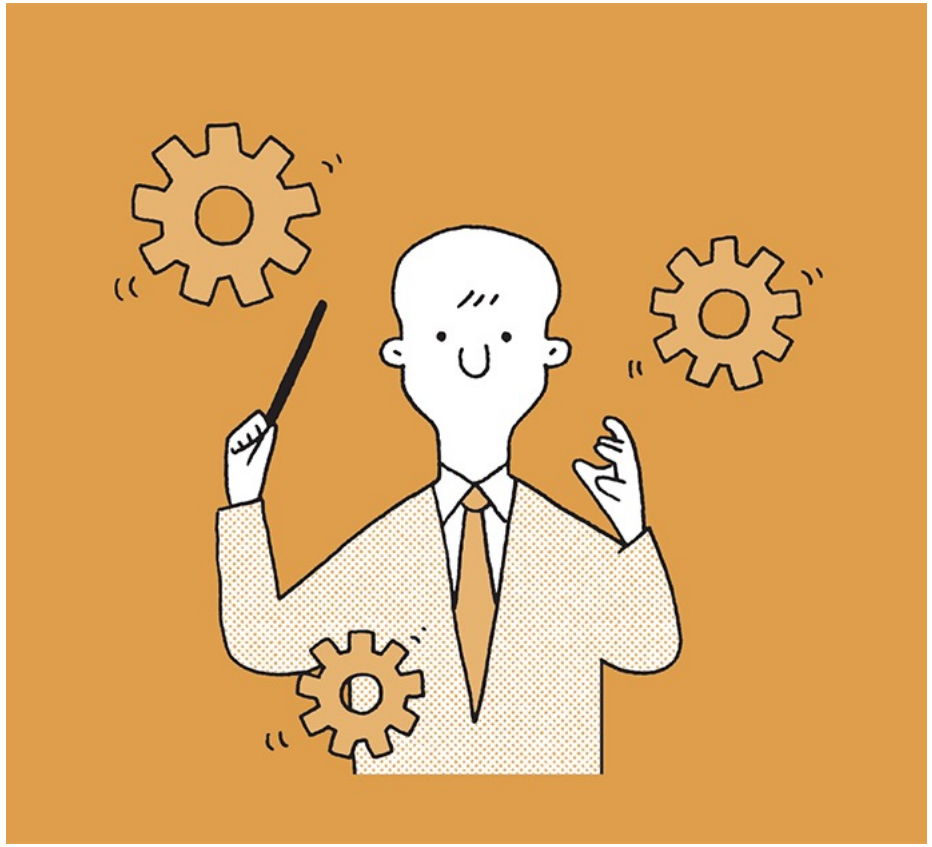
UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff

Second step is to run the model-agnostic processes

DALLE generated image

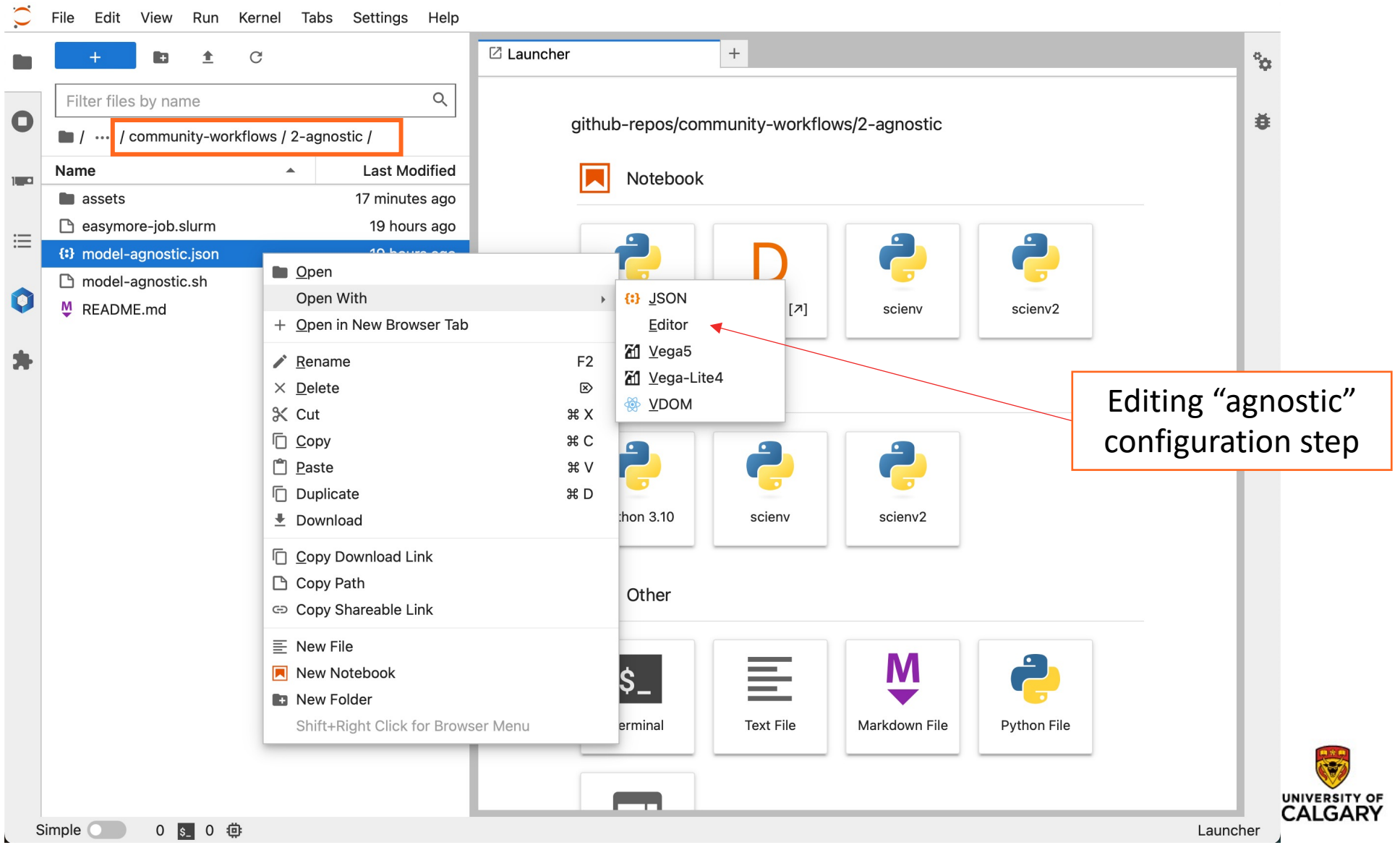
Practice Example – Bow River at Banff



- Through a tool called “agnostic orchestrator” all the relevant data needed for this setup is processed by the Graham HPC
- We have previously downloaded all the relevant tools for the “agnostic” step
- We use the “agnostic orchestrator” configuration file instruct the file processing hierarchy

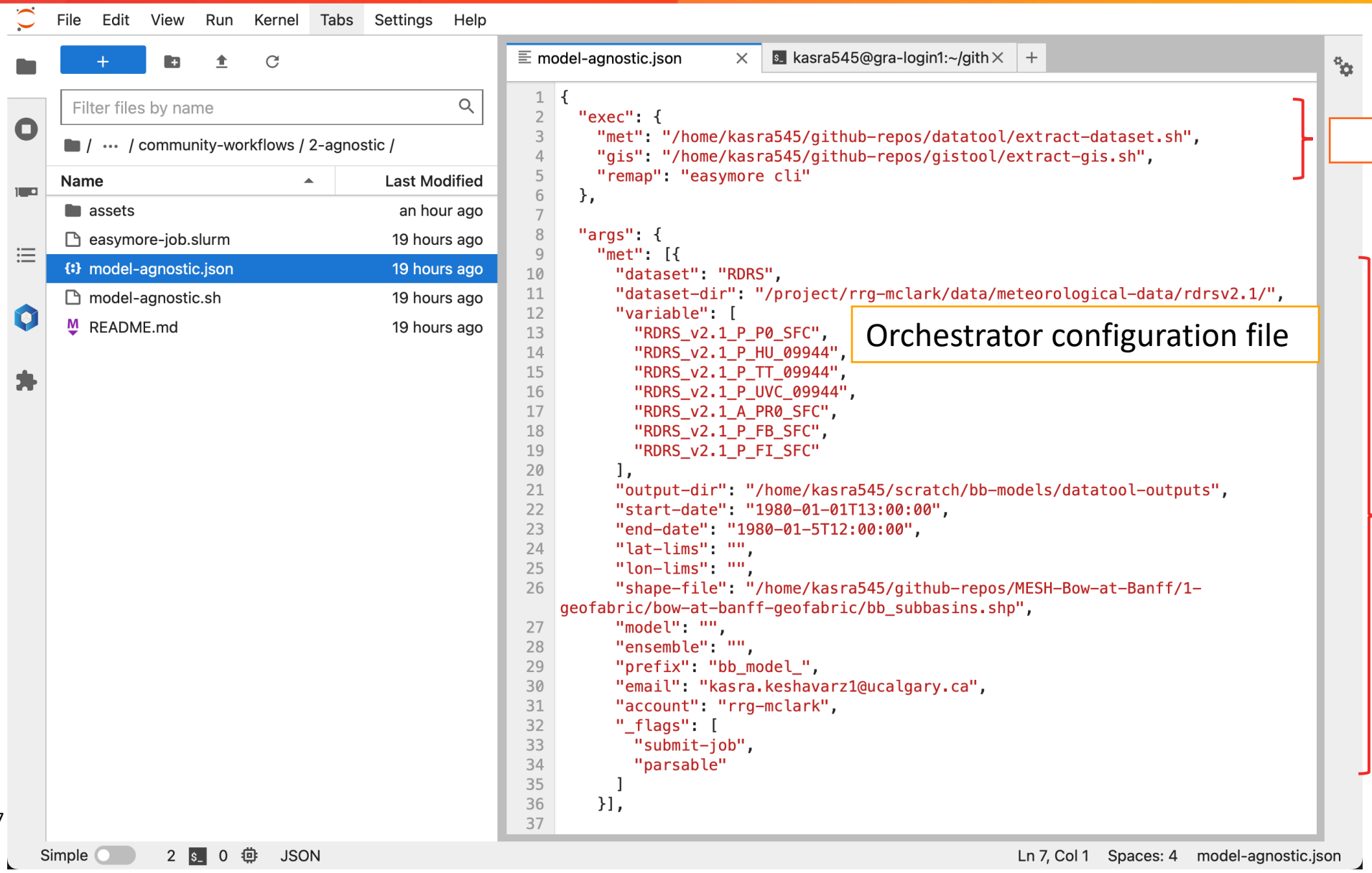
Practice Example – Bow River at Banff

Agnostic workflow execution using its “Orchestrator”



UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff



Executable paths

Orchestrator configuration file

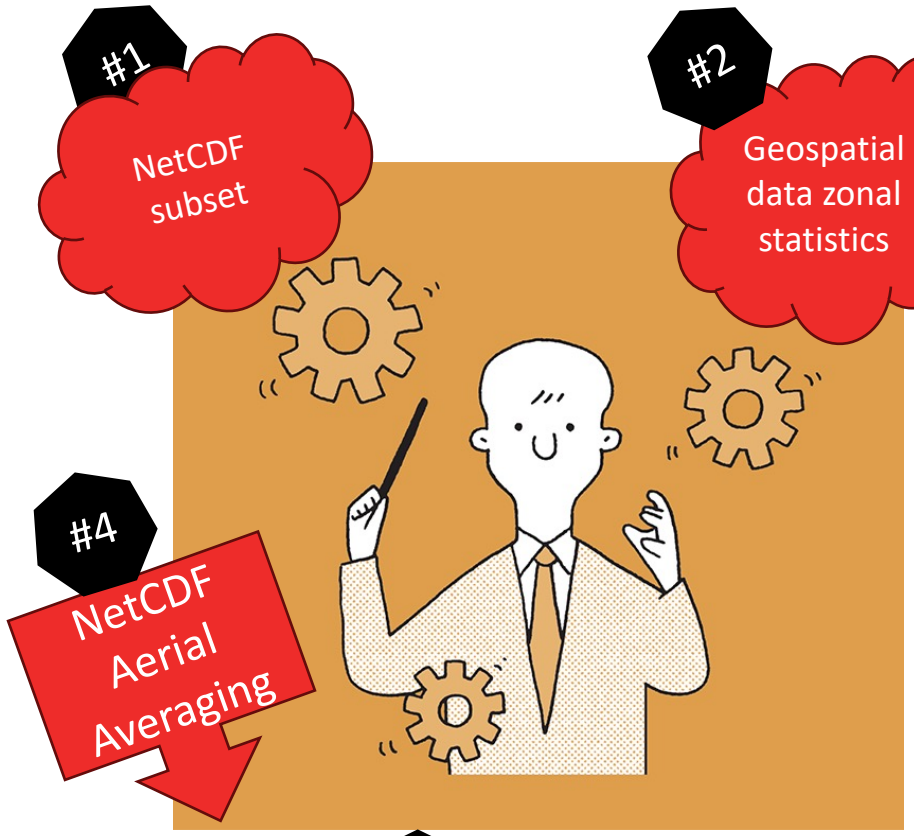
Executable arguments and options

(empty options are ignored)

This is the only file for the “agnostic” step

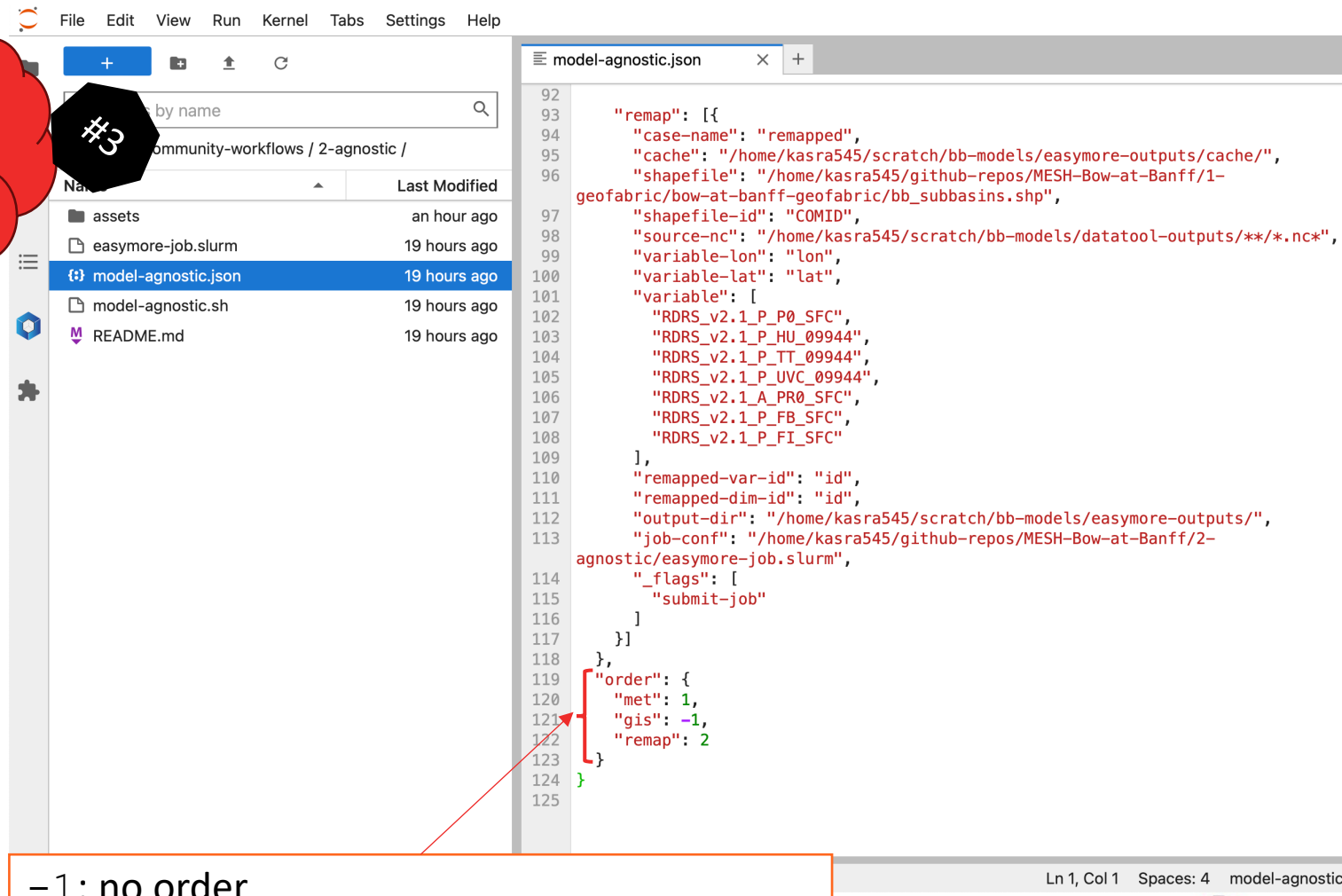
UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff



: Order of process

Automated and ordered job submission to HPC



`-1`: no order

`ordered integers`: order of processes

UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff

Switching to the JupyterLab webpage

for a quick overview

Execute the “agnostic orchestrator” in the Terminal using:

```
$ ./model-agnostic.sh model-agnostic.json
```

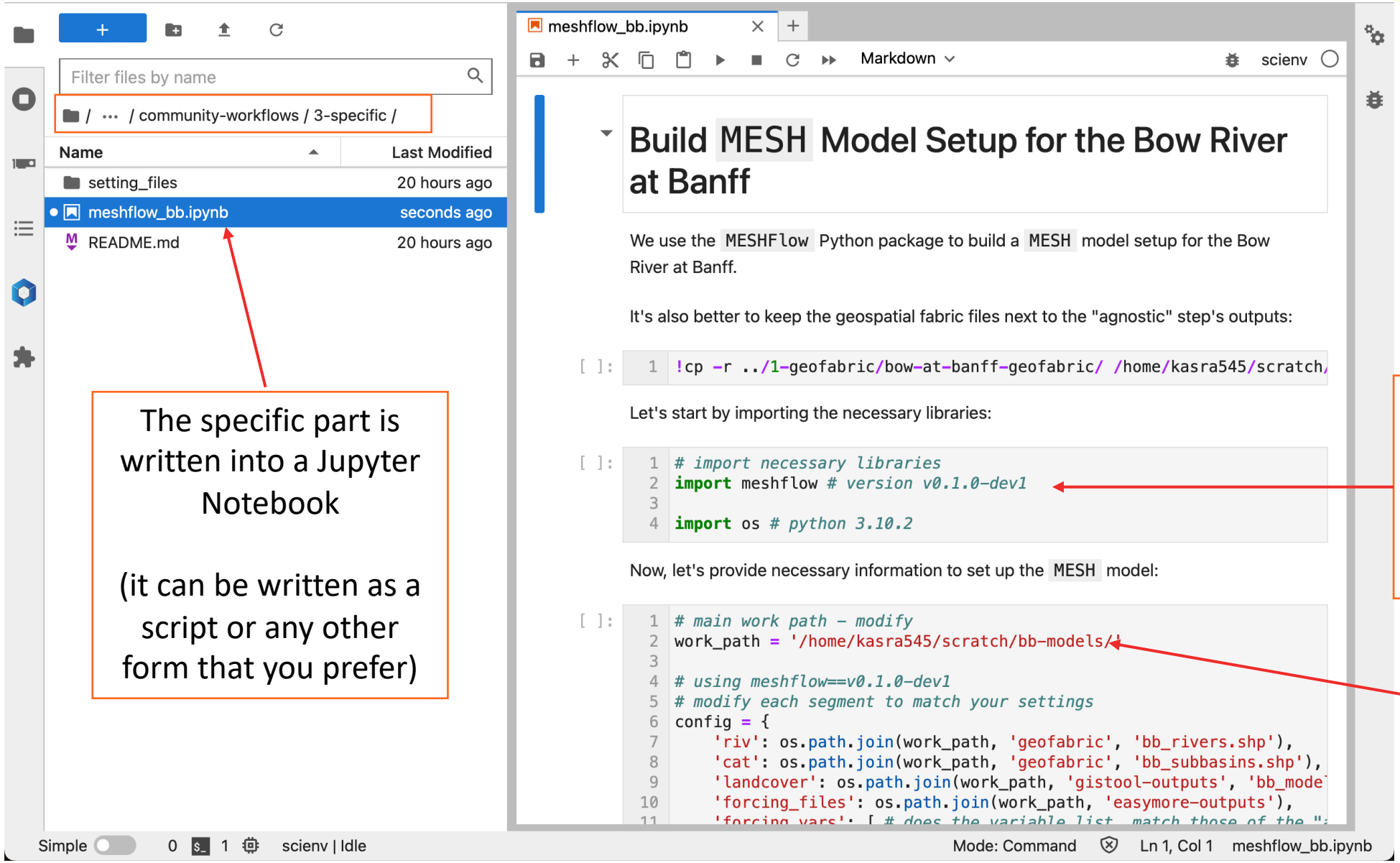
Any questions so far?

Practice Example – Bow River at Banff

Third and final step is to run the model-specific processes

DALLE generated image

Practice Example – Bow River at Banff



The specific part is written into a Jupyter Notebook

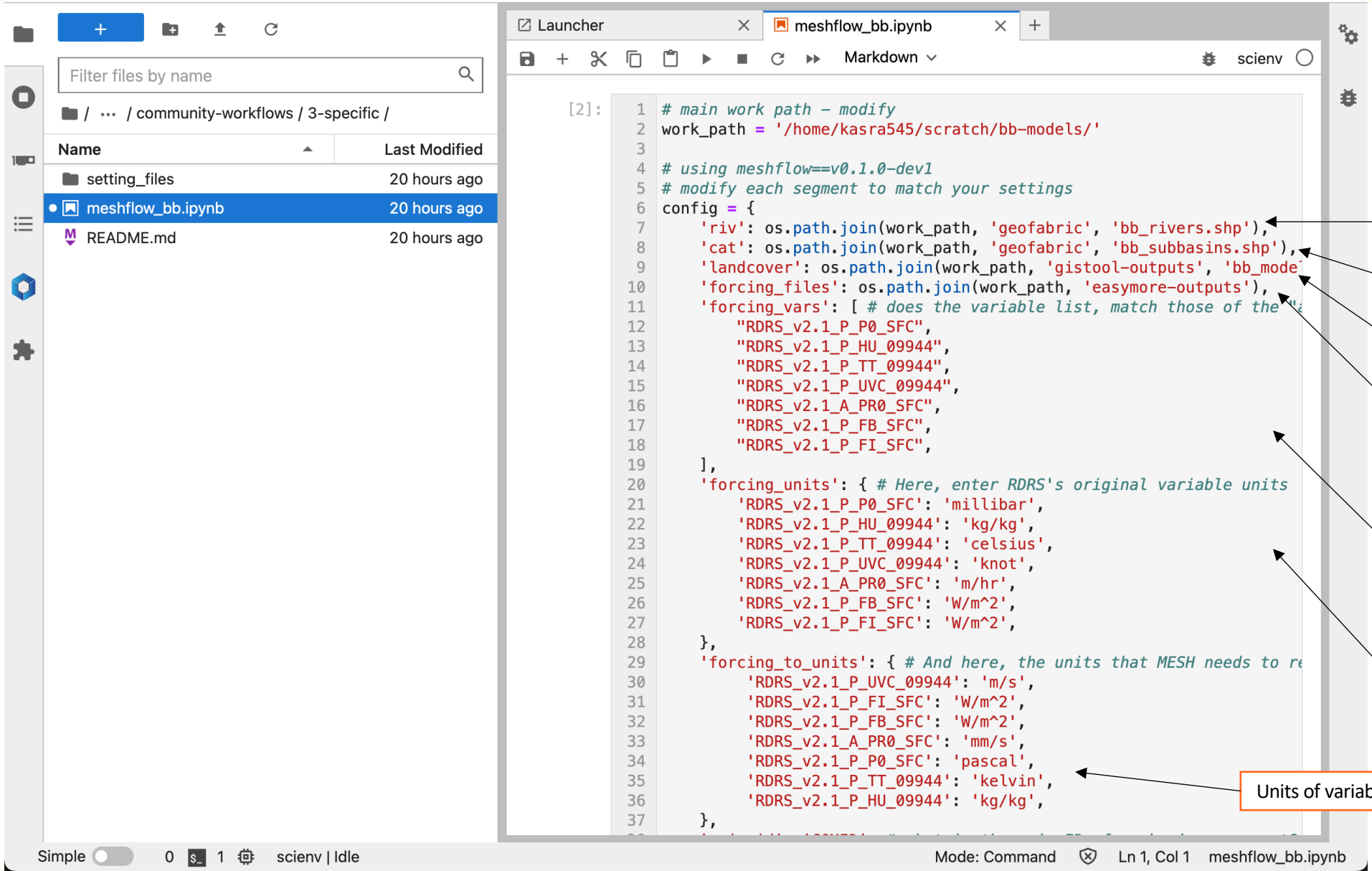
(it can be written as a script or any other form that you prefer)

`Meshflow` is the package to build a MESH model out of all the data we processed so far

The root address of where the data is stored

UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff



River network geometry file

Subbasin geometry file

Landcover fractions file(s)

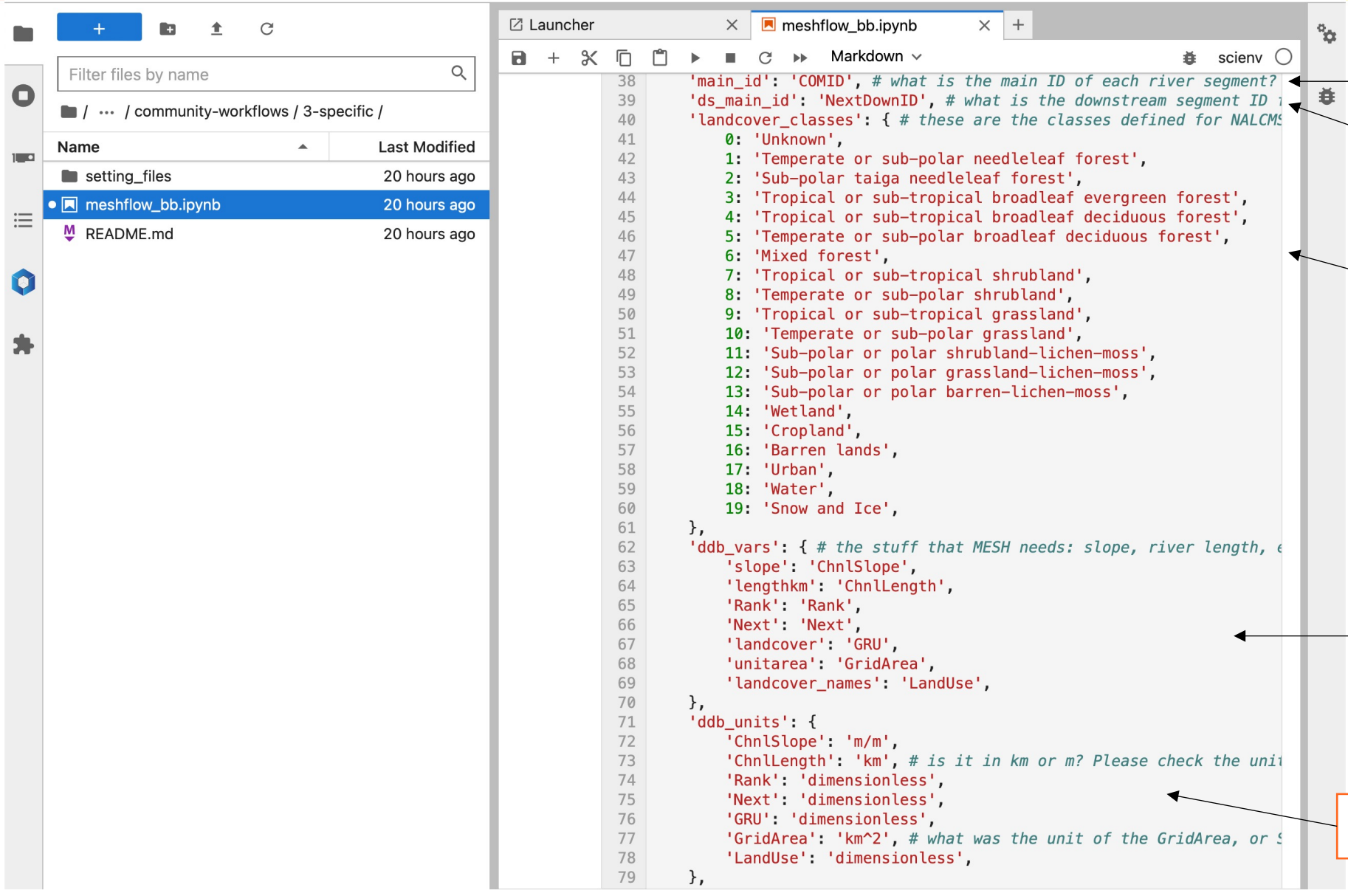
Prepared forcing file(s)

List of variables to be used

Variables' default units in the NetCDF files

Units of variables that MESH requires

Practice Example – Bow River at Banff



River Segment/Subbasin ID

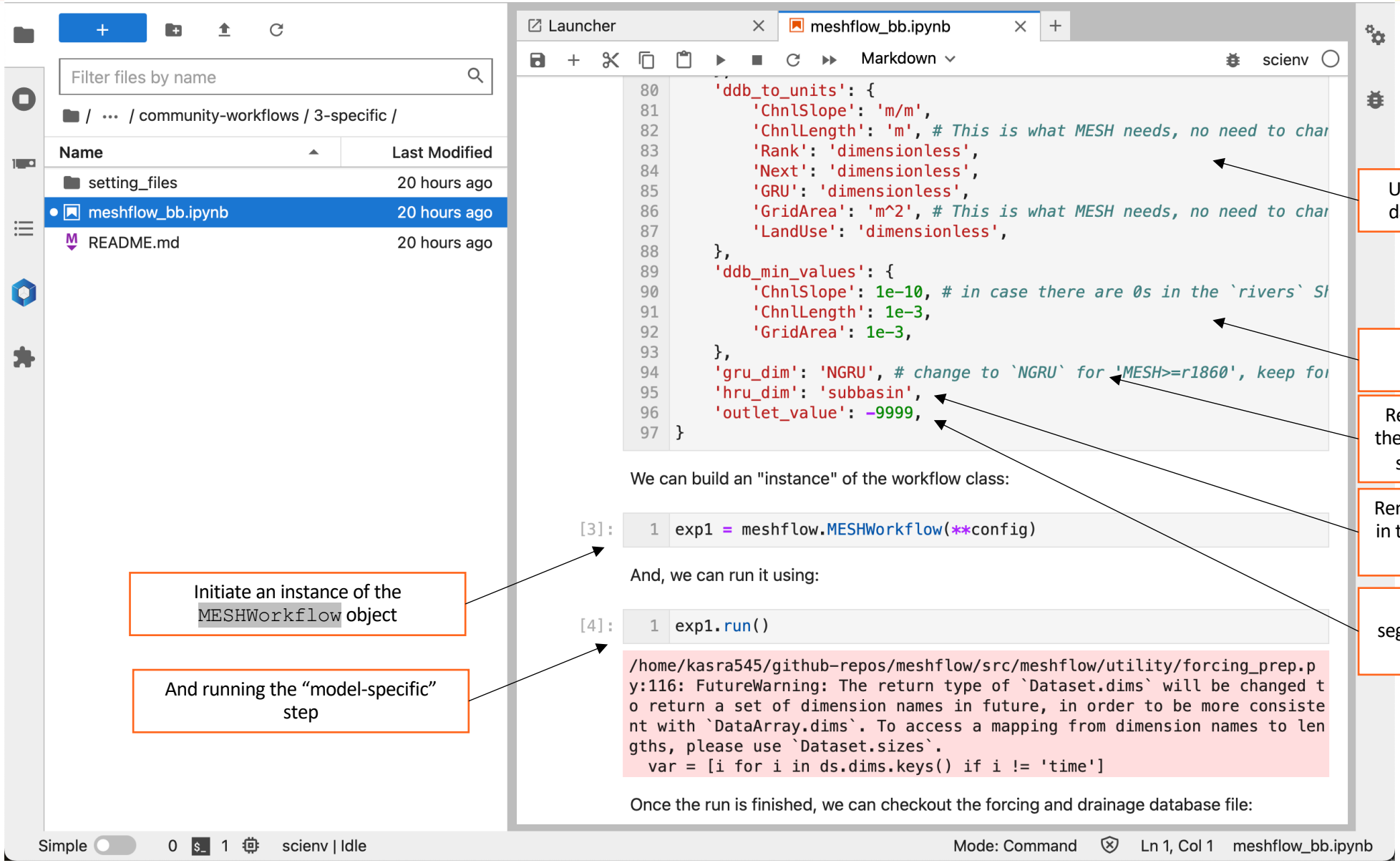
Downstream river segment/subbasin ID

Landcover classes of the landcover dataset used

All the variables to be used in the “drainage database” MESH file

Units of variables to be included in the “drainage database” MESH file

Practice Example – Bow River at Banff



Units of variables in the “drainage database” file that MESH requires

Minimum values of “drainage database” MESH file

Renaming GRU dimension name in the MESH “drainage database” file to satisfy diverse versions of MESH

Renaming sub-basin dimension name in the MESH “drainage database” file if needed

The value of the outlet river segments created using “`Hydrant`” (`Hydrant` default is `-9999`)

Initiate an instance of the `MESHWorkflow` object

And running the “model-specific” step

UNIVERSITY OF CALGARY

Practice Example – Bow River at Banff

Switching to the JupyterLab webpage

for a quick overview

Any questions so far?

Outline

- Community Hydrological Modelling Recipe
- Digital Infrastructure on Digital Research Alliance of Canada's HPCs
- Setting Up Your Account on HPCs
- Introducing Community Workflows & Tools
- Practice Example – Bow River At Banff
- Troubleshooting

Troubleshooting

Whenever you face an issue, follow the steps below:

- “Agnostic” log files are located in the following directories:

```
$HOME/.datatool
$HOME/.gistool
/path/to/EASYMORE/s/temp/directory
/path/to/agnostic/orchestrator/s/directory
```

Look for files with the following extensions:

```
.out, .err, .log
```

Try to understand what has gone wrong. In 80% of the time, there is a minor human error involved.

Troubleshooting

Continuing…

- Make sure Graham storages are responsive, very often than not, the `home`, `scratch`, or `project` spaces experience technical difficulties
- Make sure you have enough space on any storage you are working; the `quota` command can show you how much storage you are using
- If the `project` space gets full, communicate this with your PI; users do mistakenly fill up the shared storage without much justification,
- Try troubleshooting technical issues using ChatGPT or any online ChatBot you have access to

Troubleshooting

- Try troubleshooting by Google-ing and searching Stackoverflow
- Avoid using chat mechanisms to exchange critical information; everything needs to be documented properly; undocumented work equals to null
- If nothing worked out, open an “Issue” on the relevant GitHub repository(-ies) providing an MCVE
- Feel free to contribute to any repository you are interested in. Whenever you `commit`, follow the `commit` guidelines recommended by the creator of `Git`
- Fix `MESH` bugs

Troubleshooting

Linus Torvalds – creator of Linux and Git

```
Header line: explain the commit in one line (use the imperative)

Body of commit message is a few lines of text, explaining things
in more detail, possibly giving some background about the issue
being fixed, etc.

The body of the commit message can be several paragraphs, and
please do proper word-wrap and keep columns shorter than about
74 characters or so. That way "git log" will show things
nicely even when it's indented.

Make sure you explain your solution and why you're doing what you're
doing, as opposed to describing what you're doing. Reviewers and your
future self can read the patch, but might not understand why a
particular solution was implemented.

Reported-by: whoever-reported-it
Signed-off-by: Your Name <youremail@yourhost.com>
```

Same philosophy applies to GitHub Issues

Thank you for attending

https://ucalgary.ca/civil

UNIVERSITY OF CALGARY