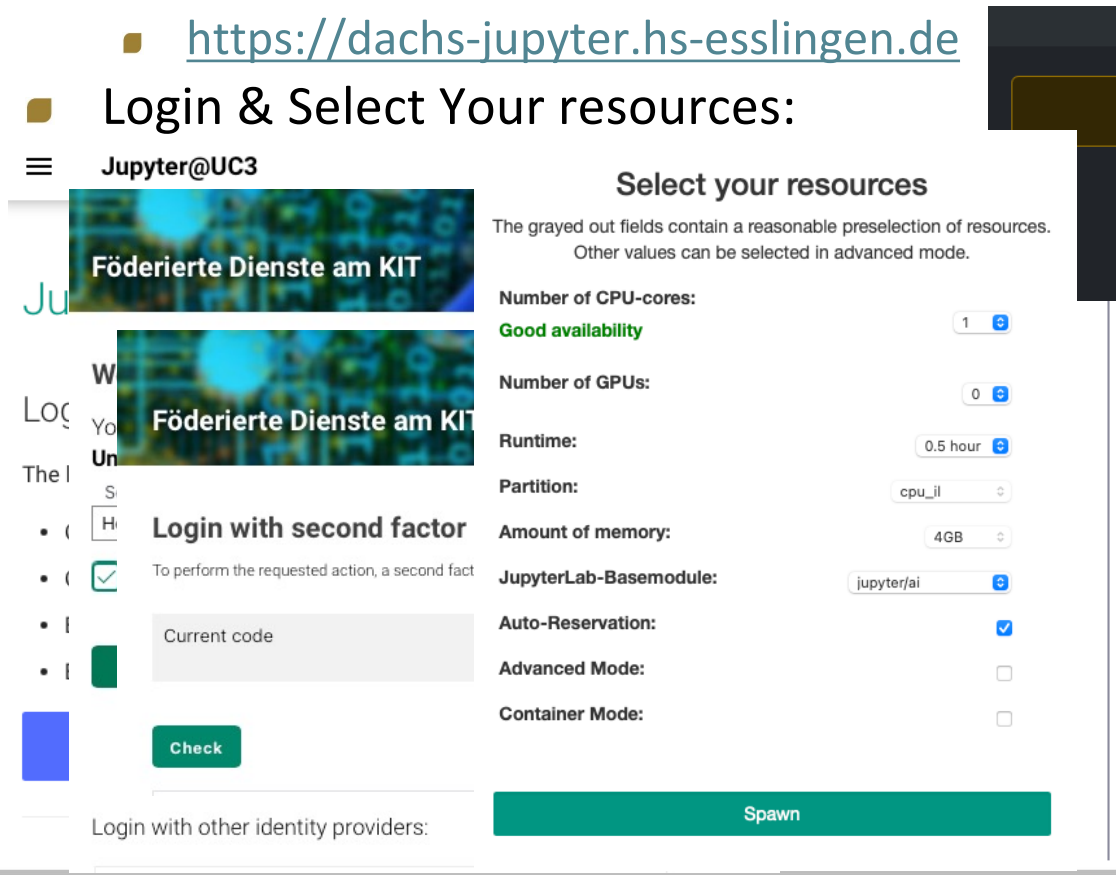
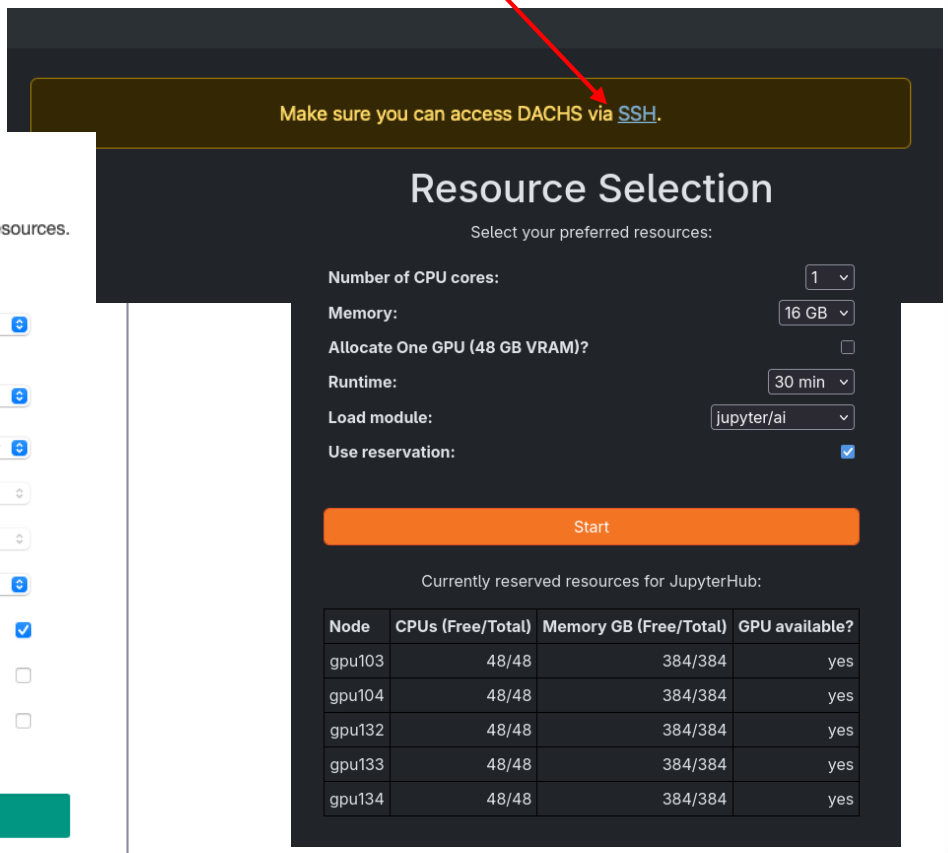
- JupyterHub on bwUniCluster & DACHS

JupyterHub on bwUniCluster 3.0 and DACHS 1/2

- JupyterHub is a Web-frontend to Jupyter Notebooks (**Ju**lia, **Pyt**hon & **R**) providing *interactive* Python w/ formatting in a Web-Browser (saved as JSON)
- With an account – and having logged in once to create Your home – use:
  - https://uc3-jupyter.scc.kit.edu or
  - https://dachs-jupyter.hs-esslingen.de
- Login & Select Your resources:





JupyterHub on bwUniCluster 3.0 and DACHS 2/2

A word about resources:

- This submits the `jupyterhub-spawner` on Your behalf
- If You select GPU: **one whole** node for yourself, otherwise shared!

For this Workshop today, we have 20 nodes reserved for now…

- Please be considerate: at Resource selection, check # of free nodes
  Remember: *Interactive* use is the *least* efficient usage!

- If You need a number of nodes for a lecture at specific times/dates:
  Please write an email to dachs-admin@hs-esslingen.de
- Currently, there's no interactive Jupyter access to multi-GPU nodes

JupyterHub: First steps 1/3

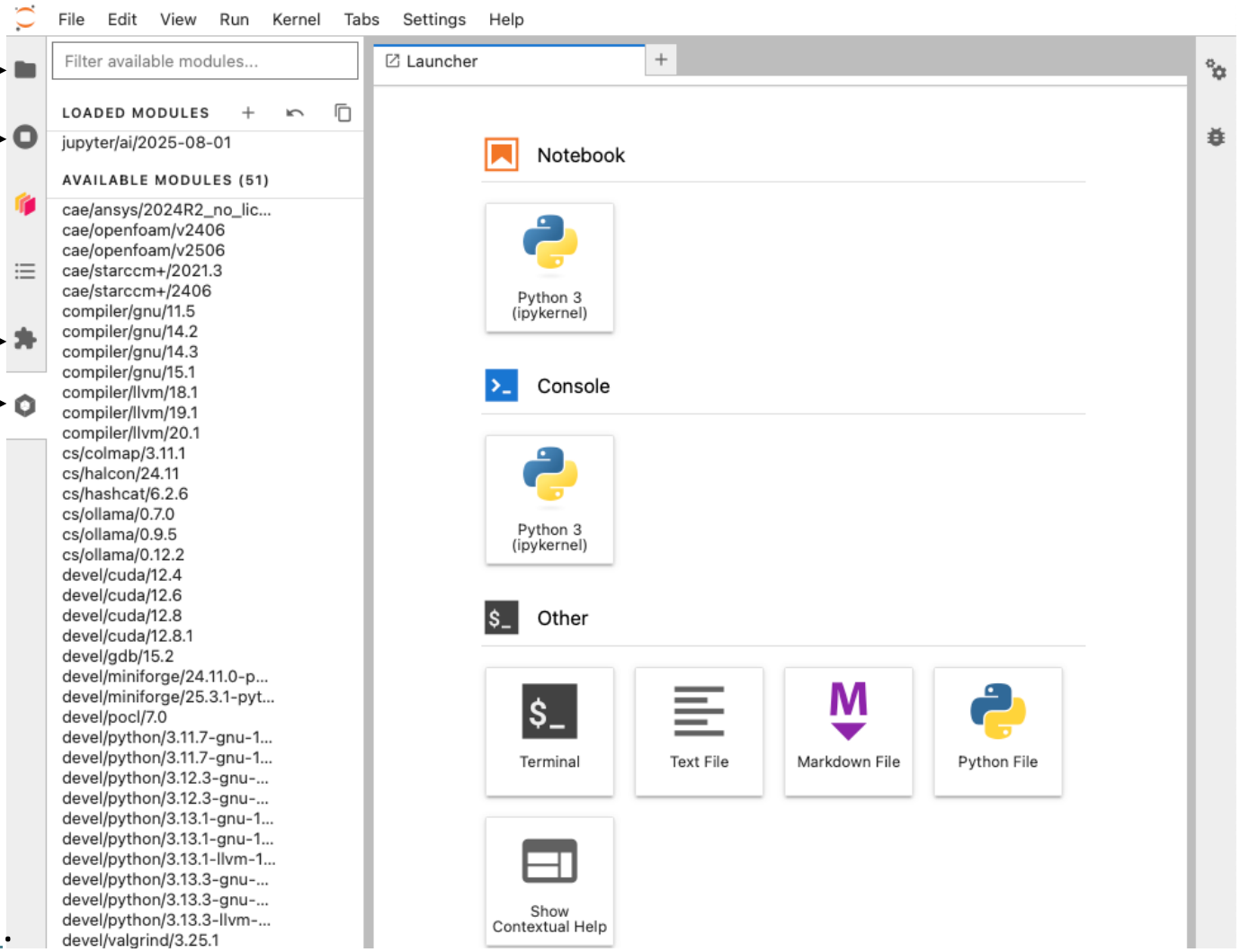
- First view (without existing Notebook `Untitled.ipynb`)

Files

Controlling Tabs

Jupyter Extensions, like `jupyter-matplotlib`

Loaded & Available Modules pane:



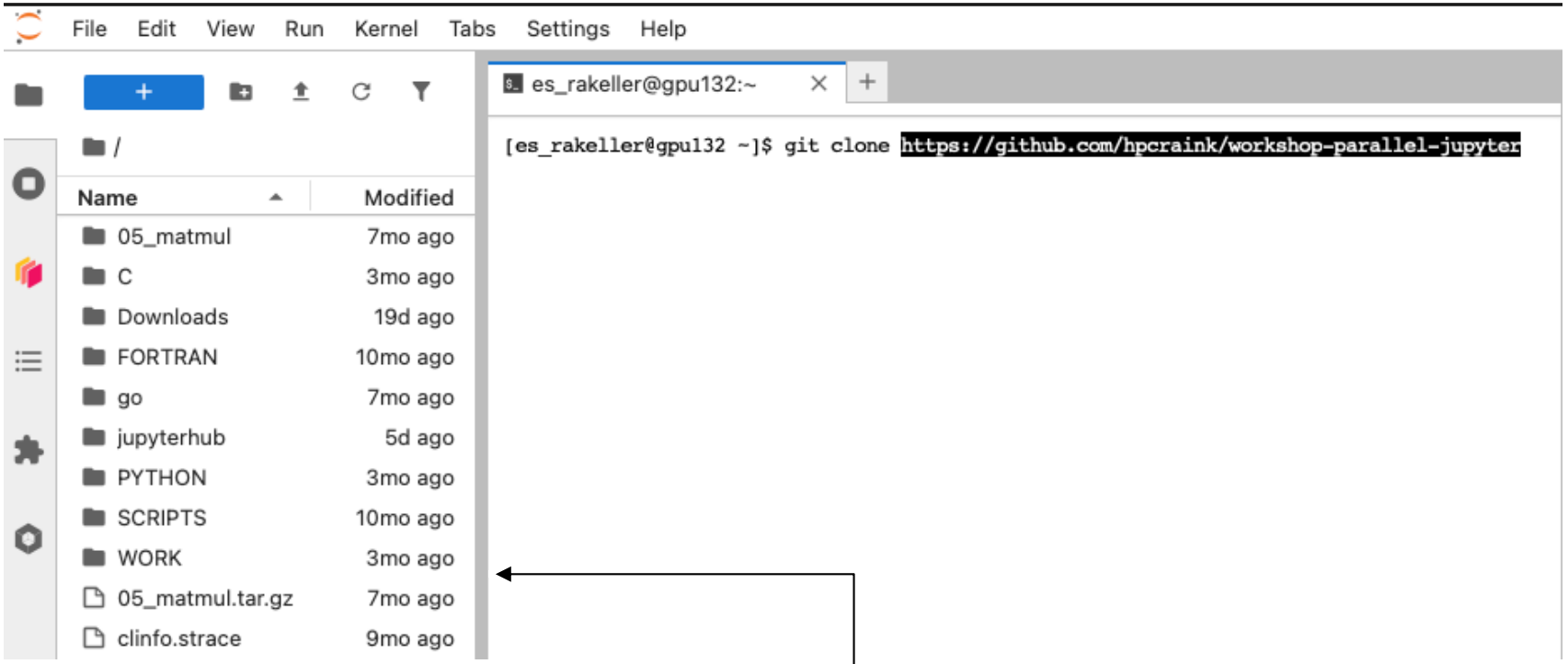
The default module `jupyter/ai` loads:

- CUDA
- Torch & Tensorflow
- Pandas
- SciKit-Learn
- Seaborn and more

For other modules: Wiki.

JupyterHub: First steps 2/3

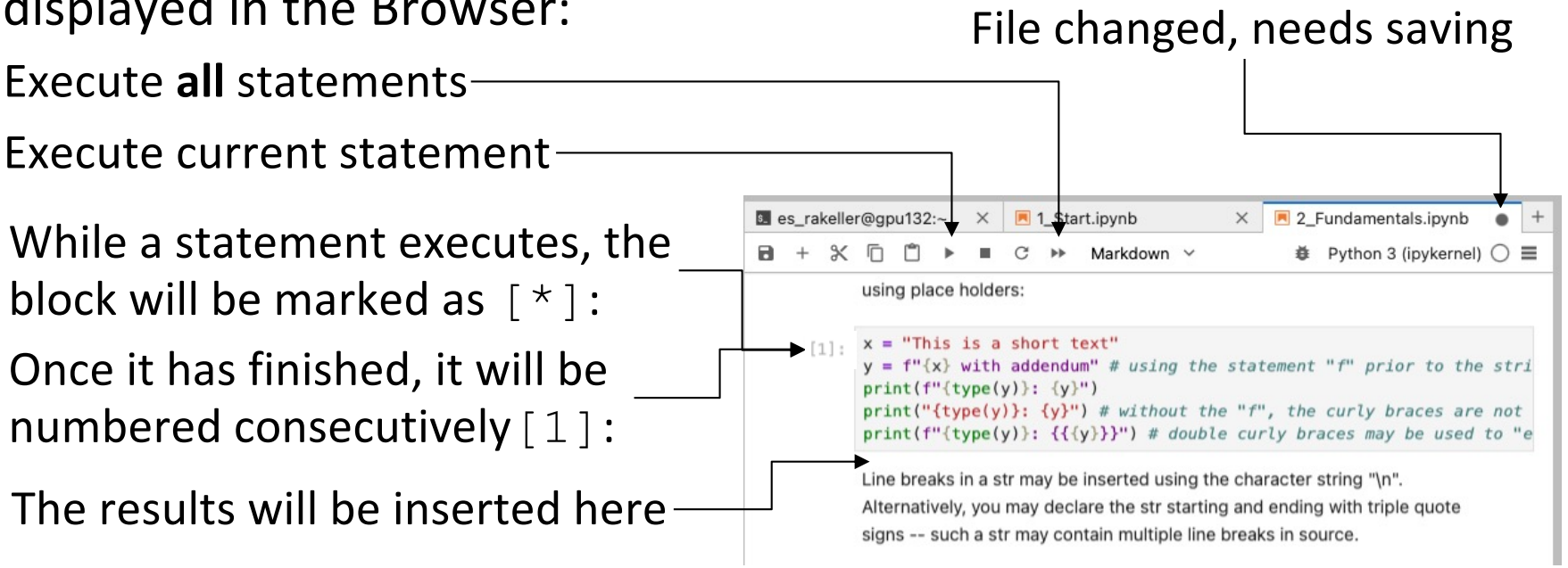
- For example for best practice (on Jupyter, Python, Visualization, …)
  `git clone https://github.com/hpcraink/workshop-parallel-jupyter`
- In a new terminal:



- It will show up **here** in a few seconds
- Then click on the cloned directory and open `1_Start.ipynb`

JupyterHub: First steps 3/3

- The Interactive Python Notebook File (`.ipynb`) contains Markup, Python code and executed results and stores visualization to be displayed in the Browser:



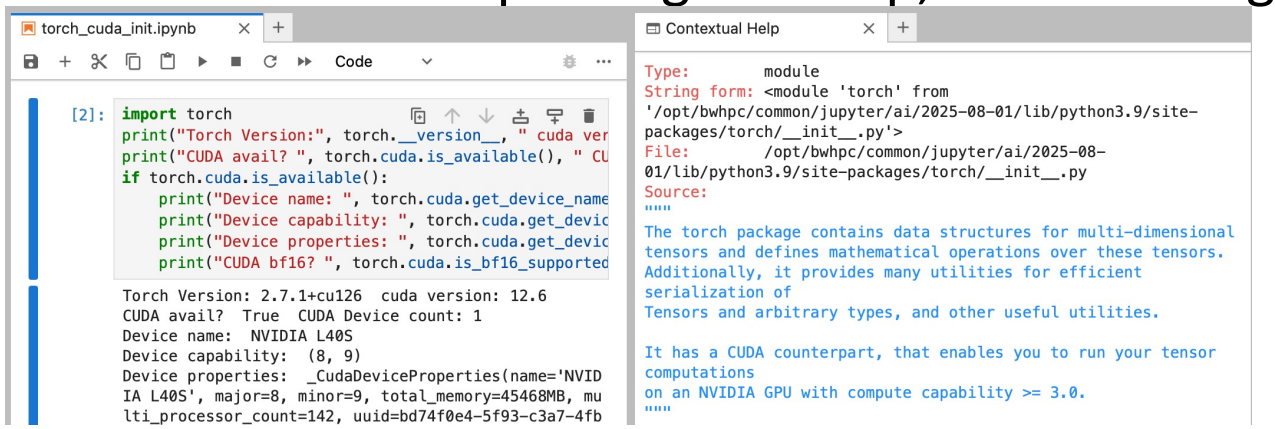
- `2_Fundamentals` explains Python programming
- `3_Numpy` explains basic Matrix operations using NumPy
- `4_Pandas` shows how Pandas works with large Parquet files
- `5_Machine_Learning` introduces SciKit Learn with Visualization

JupyterHub: Best practices

- Using a GPU node, check in Terminal the output of `nvidia-smi`...
- When using PyTorch, check the output of:

```
import torch
print("Torch Version: ", torch.__version__)
print("CUDA version: ", torch.version.cuda)
print("CUDA avail? ", torch.cuda.is_available())
if torch.cuda.is_available():
    print("Device count: ",      torch.cuda.device_count())
    print("Device name: ",       torch.cuda.get_device_name())
    print("Device capability: ", torch.cuda.get_device_capability())
    print("Device properties: ", torch.cuda.get_device_properties())
    print("CUDA bf16? ",         torch.cuda.is_bf16_supported())
```

- The Contextual Help is of great help, when editing (right mouse click)



JupyterHub: on bwUniCluster Please Stop Server

- **On bwUniCluster Prior** to logging out, end session to free resource: