JupyterHub for teaching on LUIS cluster

We provide an instance of the JupyterHub platform to support giving lectures in the LUH compute cluster environment. The JupyterHub server also runs the system NbGrader, which allows you to assign, distribute and (automatically) grade Jupyter notebooks. The JupyterHub platform is intended only for non-critical teaching activities that require LUIS cluster resources. If you want to use Jupyter notebooks for your own research, use the Open OnDemand web portal of the cluster instead.

JupyterHub is available via the URL https://jupyterhub.cluster.uni-hannover.de. If you want to connect from the “outside” (e.g. from Home), you'll have to establish a VPN connection to the university’s network first. See VPN Service for details.

Please make sure you have read and accepted the following notes for using the JupyterHub service in the LUIS compute cluster.

Usage Conditions for the JupyterHub service

Follow this link to read the German version.

Support
The JupyterHub service, which is operated as part of the LUIS "Scientific Computing" service, directly accesses the resources of the LUH computing cluster. Therefore, the rules governing the use of LUIS IT systems and services, as well as the description in the cluster wiki, apply.

Availability
The JupyterHub service is available under the same terms and conditions as the other resources of the LUH computing cluster. Continuous operation of the service cannot be guaranteed.

Usage
The JupyterHub service is offered solely as a teaching and learning tool. The use of JupyterHub for exams is **expressly prohibited** .

How to register and manage courses

To get started with the JupyterHub platform, you will need to create cluster accounts using the BIAS system. Note that BIAS has additionaly functionality to create bulk cluster accounts. Talk to the BIAS team for further instructions. Accounts are required for both the instructors of a course and the students taking the course.

Note: We ask you to delete (using the BIAS system) student accounts that are no longer needed after completing the course .

Before the accounts are granted access to the platform, you will need to initially register your course on JupyterHub. To do this, please contact the cluster team at cluster-help@luis.uni-hannover.de and provide the following information:

- Course name (the name should only contain unicode letters, digits, spaces and the characters `_.-,`)
- Cluster username/account (and optionally first name, last name and email address) of at least one instructor of the course

A course may have multiple instructors, each with their own account/username. Each registered course receives a unique identifier in the form of the string `course<some number of digits>`.

After you receive notification that initial registration of the course is done, the course instructors can log into the JupyterHub platform at https://jupyterhub.cluster.uni-hannover.de (see figure 1)

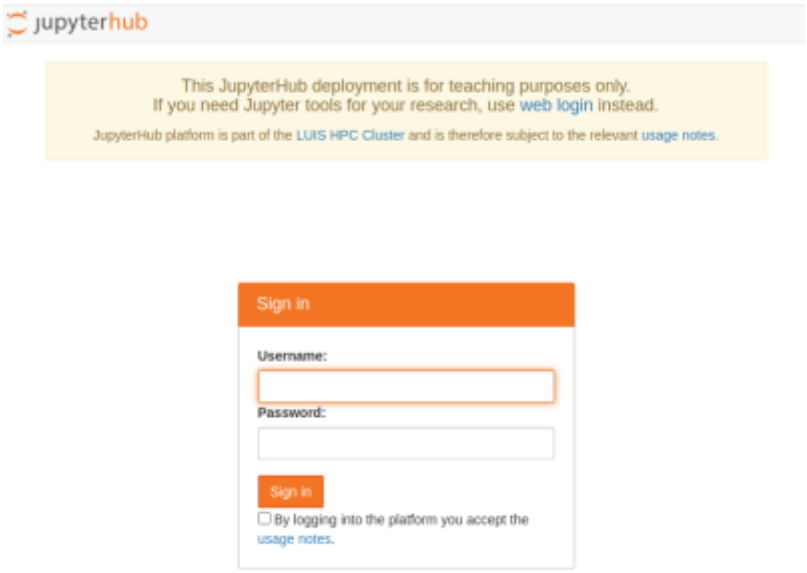

Fig. 1: JupyterHub login page

and access the registered courses from the menu **`Available Courses`**, see figure 3. To get to the menu page, click the button **`To Available Courses`** after logging in, figure 2.

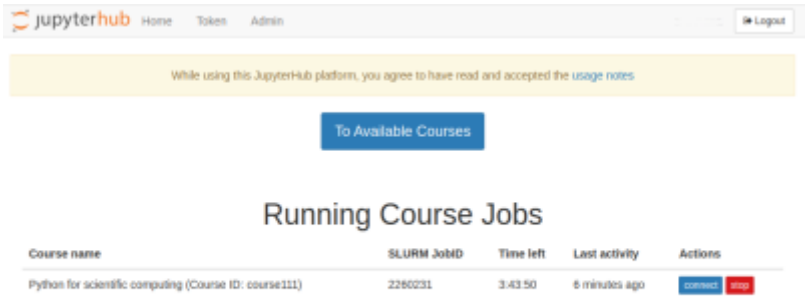

Fig. 2: JupyterHub home page

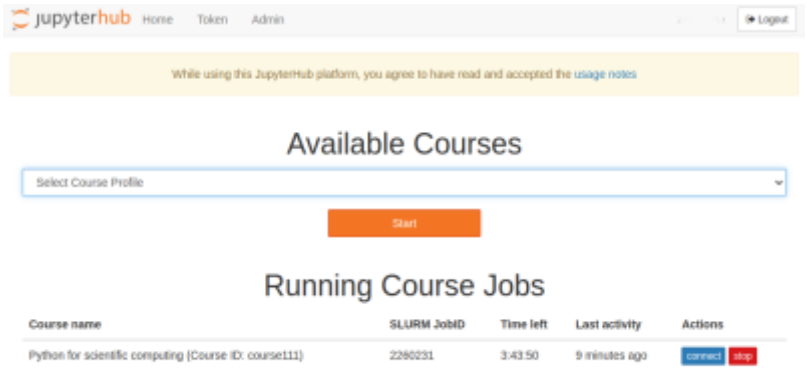

Fig. 3: JupyterHub courses menu page

By selecting a course from the menu, the page displays the course settings, such as SLURM options, a list of instructors and students assigned to the course, location of course data in the cluster, etc. figure 4. The information shown varies depending on whether the username is registered as an instructor or student with the selected course.

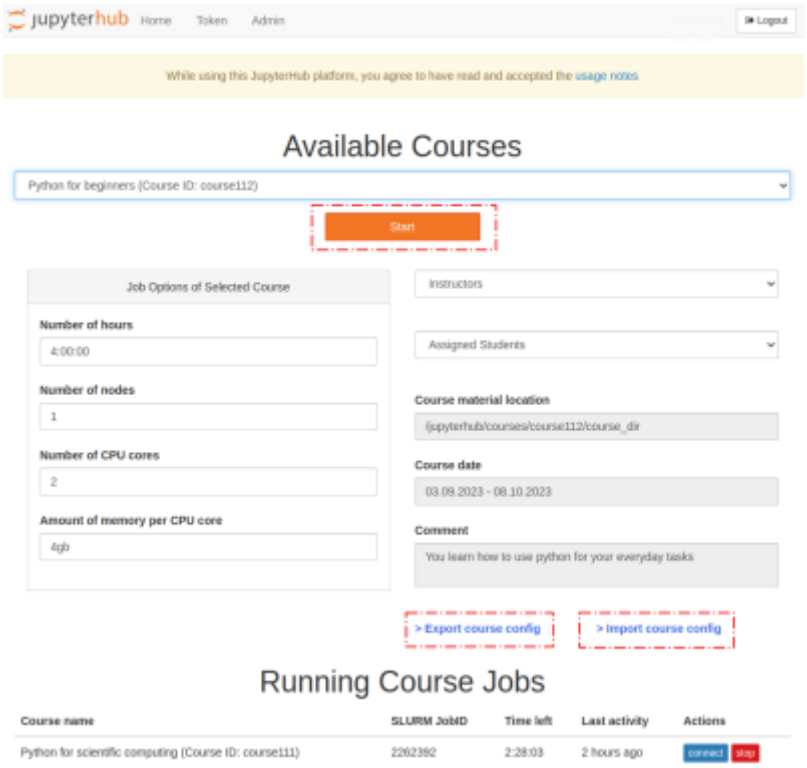

Fig. 4: Settings of the selected course

To start a Jupyter notebook for the selected course, click the link **`Start`**, see figure 4. This submits a SLURM batch job to the cluster. The page opened will show the state of the job in the queue including the ID of the job, which is underlined in red in figure 5. The time a course job waits in the queue depends on the job requirements and the current workload of the cluster.

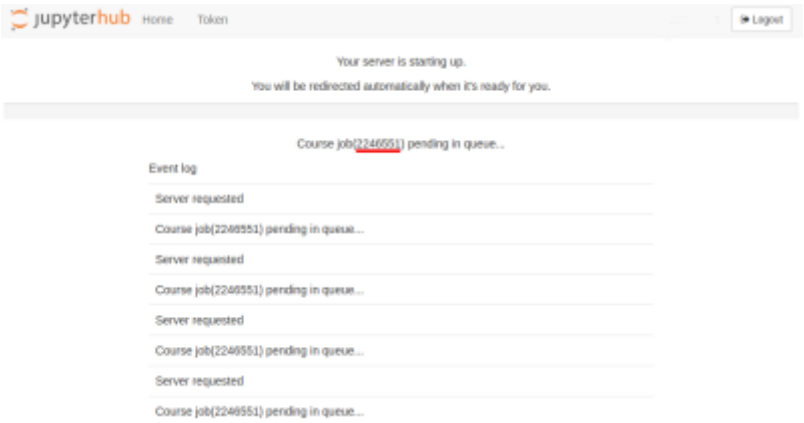

Fig. 5: Course job in queue

Note: if a course job fails and you need support, please include the job ID in your request, see figure 5 . However, since standard output and error streams of the job are directed to the `~/jupyterhub_<course id>_<job id>.log` file in your $HOME directory, you may also examine the job log content yourself.

Note: If you intend to present in class, we may (on request) temporarily reserve a some amount of cluster resources to reduce your wait times. However, it is not possible to regularly reserve a large amount of nodes. To arrange for this, please provide us with the required amount of resources (number of CPUs, memory, walltime and other options per course job), the probable number of course participants as well as the start and end dates of the lectures several weeks (at least three or four weeks in advance would be good) before the course starts. As you may already know, we offer a so-called FCH service, which allows institutes to have their own nodes integrated into the LUIS cluster to get additional and somewhat exclusive resources. These nodes can easily be reserved for your teaching activities as long as the maximum general reservation of 60 hours/week is not exceeded, whereas we'll need to strike a balance for the generic resources that largely have to remain available for the whole LUH cluster community.

To manage your registered courses, use the links **`> Export course config`** and **`> Import course config`** on the course menu page, figure 4. The link **`> Export course config`** exports the current configuration of the selected course as a JSON file. The file name must be the course ID with the extension `json`. Open the JSON file and make appropriate modifications. To activate the changes, use the link **`> Import course config`**, which uploads the JSON file to the JupyterHub server. Note that the JSON file name must match the ID of the course whose configuration you want to change. The following course options can be modified:

- Course state
- List of instructors
- List of students
- Course SLURM job options (long form of the options must be entered)
- The name and location of the Conda environment used for the course
- Course start and end dates
- Course comment

The following in an example JSON configuration for the course with the ID `course010` (save the lines to the file `course010.json`. Names are random):

course010.json

```
{
```

```
  "state": "active",
  "name": "Python for scientific computing",
  "comment": "Python for scientific computing",
  "teachers": [
    "username1,John,Chase,j.chase@uni-hannover.de",
    "username2,Juliet,Williams,williams@uni-hannover.de"
  ],
  "students": [
    "stud0001,Armaan,Nunez,nunez@stud.uni-hannover.de",
    "stud0002,Angelica,Vargas,angelica.vargas@stud.uni-hannover.de",
    "stud0003,Jago,Clements,clements@stud.uni-hannover.de",
    "stud0004,Suzanne,Barker,barker@stud.uni-hannover.de"
  ],
  "location": "/jupyterhub/courses/course010/course_dir",
  "start": "03.09.2023",
  "end": "08.12.2023",
  "job_options": " --time=04:00:00 --nodes=1 --cpus-per-task=4 --mem-
per-cpu=4G --gres=gpu:1",
  "jupyter_notebook": {
    "autosave_interval": 10,
    "kernel_default": {
      "name": "Python Computing 2023",
      "conda_env":
"/jupyterhub/courses/course010/conda_envs/python_computing_2023"
    }
  }
}
```

The section "jupyter_notebook" can be removed from the configuration, as the parameters defined there ("autosave_interval", "kernel_default") are optional.

Each student and instructor entry above is a comma-separated string of the following items: "<username>,<first name>,<last name>,<email address>". Where items are ordered, all of them except <username> are optional.

The course data directory (the parameter "location" in JSON file) is located at `/jupyterhub/courses/<course id>/course_dir` in the cluster, with `<course id>` being replaced by the corresponding course ID. The data directory is where the NbGrader database and config files, Jupyter notebooks and other working directories are placed. This path is fixed and cannot be modified.

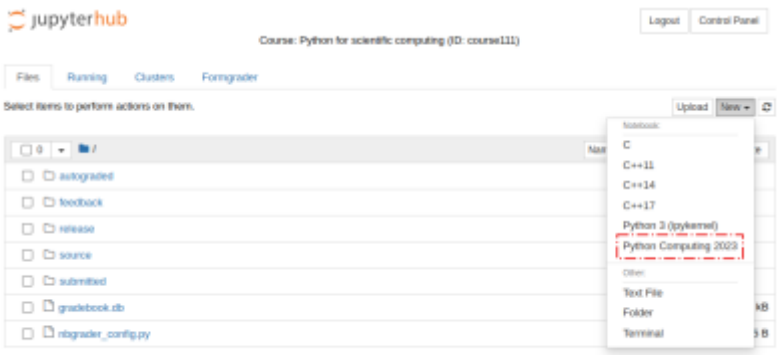

Fig. 6: Conda env. as jupyter kernel

The path to the course conda environment (the parameter "conda_env") should be prefixed with `/jupyterhub/courses/<course id>/conda_envs`. See the example configuration above. Path permissions are automatically set so that course instructors can read, create and modify, whereas enrolled students can read underlying files and directories. The conda environment assigned to a course will be accessible from the **New** menu of the Jupyter interface as a Jupyter kernel, the name of which is determined by the "name" parameter in the conda section of the course configuration, see figure 6. Before starting a course job, make sure that the conda environment of the course is already created, otherwise the job will fail. See conda usage instructions in the cluster for details.

Instructors can also access their course locations from their home directory in the cluster as `~/MY_COURSE/<course name>`.

By default, the following SLURM options are set for a course job: "job_options": " --time=04:00:00 --nodes=1 --cpus-per-task=1 --mem-per-cpu=2G"

Note: The username can be registered in either the student list or the instructor list, but not in both simultaneously.

Note: Before uploading a configuration with new usernames to the JupyterHub server, make sure that the usernames exist in the system, i.e. they must be able to log in to the cluster in one of the possible ways.

Note: If you want to remove all student usernames from your course config, set "students": " "

Note: The maximum number of student accounts per course is currently *technically* limited to 1000. **Attention:** This represents a *technical* limitation. However, it should not be interpreted as a guarantee that, in courses with approximately 100 or more participants, all jobs associated with JupyterHub instances will start immediately. Furthermore, it is generally not feasible to reserve resources for the interactive work of such large student groups, unless sufficient dedicated resources are provided and integrated into the cluster via the FCH service.

To deactivate a course for students, set "state": "inactive" in the course JSON file. Inactive courses do not appear in the student's course menu. By setting "state": "removed", you can make the course invisible in the course menu for both teachers and students. Note however that the course data are not removed but remain at their original location and therefore can still be accessed by instructors.To reactivate a removed course, please contact the cluster team.

Teachers receive a warning email before the course ends (default: 7 days). After the end date (the parameter "end" in JSON file), the course is marked "inactive". Associated SLURM jobs are cancelled, and course SLURM advanced reservations are deleted. While the course is inactive, teachers can

extend the end date. If the course remains inactive, it is marked "removed" after the 7-day grace period. Teachers receive email notifications about these transitions.

Once the course job starts, the Jupyter notebook server is launched on the allocated compute node. Instructors can manage (generate, release, collect, grade, etc.) assignments via the **`Formgrader`** menu in a Jupyter notebook session, see figure 7.

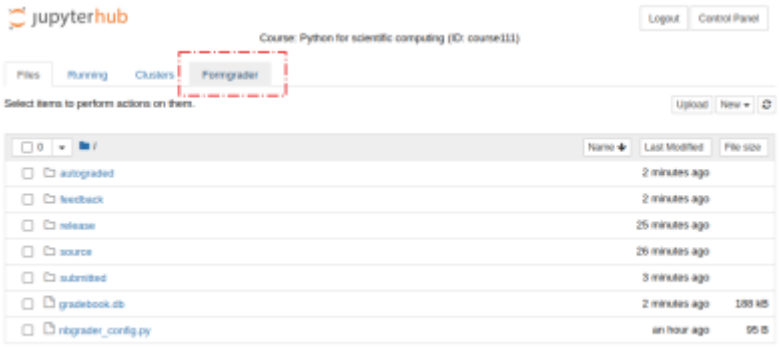

Fig. 7: NbGrader for instructors

Using the **`Assignments`** menu, figure 8, in a Jupyter notebook session, students can retrieve and submit their course assignments.



Fig. 8: NbGrader for students

Currently running course jobs are shown at the bottom of the main page, see section **`Running Course Jobs`** in figure 2 or figure 3. You can connect to the running Jupyter session using the link **`connect`**. The link **`stop`** terminates a Jupyter notebook session (i.e. cancels the corresponding SLURM job).

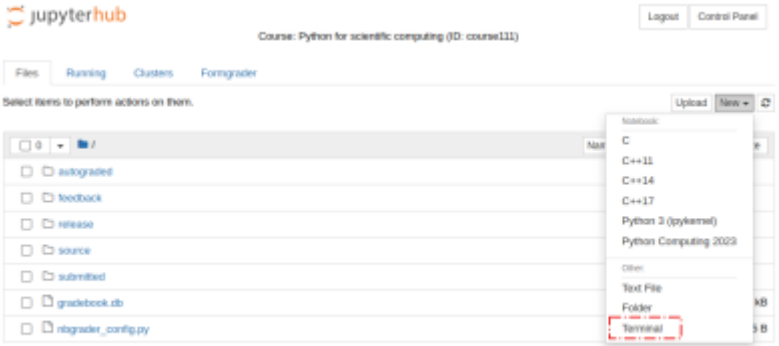

Fig. 9: Jupyer notebook terminal

NbGrader supports managing course assignments both interactively via a Jupyter notebook session as well as using the command line tool nbgrader. To get a command line, open the **`Terminal`** from the **New** menu of your jupyter session, see figure 9. Once you open the terminal, you will be taken to the course directory, where you may start working with the tool immediately. The `nbgrader` command can also be executed from anywhere because the terminal environment is configured for the current course from which it was launched.

Details on how to create Jupyter notebooks for course assignments in NbGrader format and how to use NbGrader interfaces can be found here.

FAQ

- **My JupyterHub course does not start or ends with undefined errors. What are the possible reasons?** Please check the file system quotas of your account - for example via SSH access to the cluster. If you have exceeded your limits, this would be one reason for the errors when starting the course.

From:
https://docs.cluster.uni-hannover.de/ - **Cluster Docs**

Permanent link:
https://docs.cluster.uni-hannover.de/doku.php/resources/jhub

Last update: **2026/11/09 10:21**