Getting Started with the HPC Cluster

Overview

This exercise aims to get you started and familiarize you with working on the university's high-performance computing cluster. For supplementary instructions on accessing the cluster, refer to the slide deck Working wih the HPC cluster.

Step 1: Access the HPC cluster

Firstly, ensure access to the HPC cluster via the dedicated web frontend. The cluster is accessible only from within the university network. If off-campus, you need to establish a connection via a VPN. Refer to this wiki for VPN client setup.

Once within the university network (on-campus or via VPN), you can access the HPC cluster's web frontend at https://web.hpc.hs-osnabrueck.de. Log in using your university credentials.

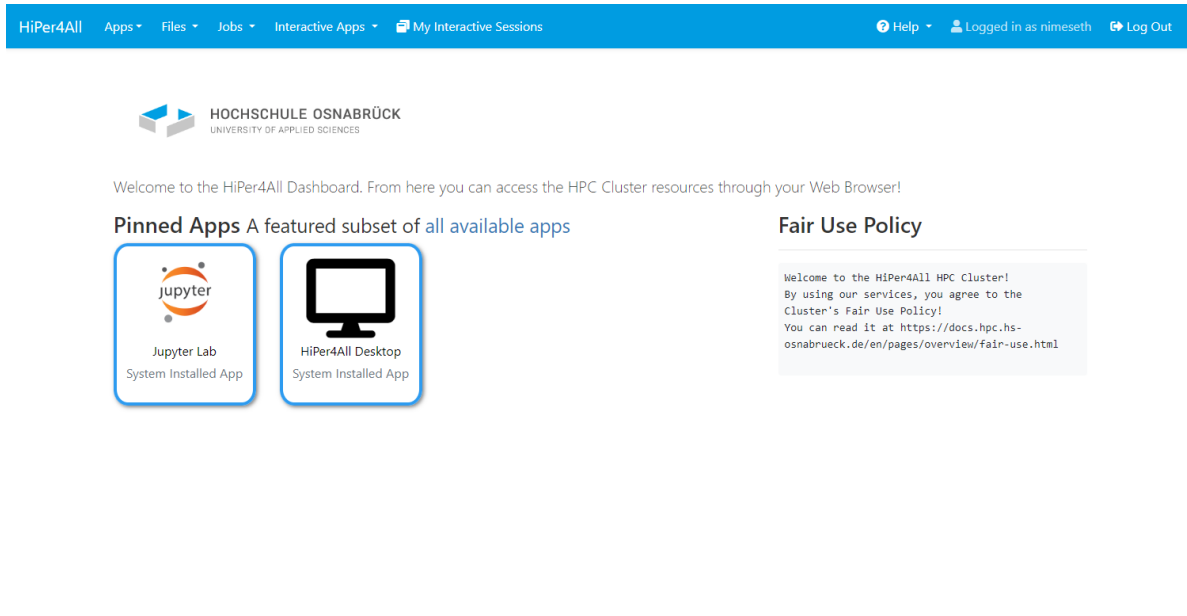You successfully completed the first step when you see the HPC cluster's web frontend as shown in the image below.

Figure 1: The HPC cluster's web frontend

Step 2: Open a SSH shell

There are various ways to interact with the HPC cluster. For executing code, we'll use Jupyter notebooks. However, before we can do that, we must ensure a functional Python environment accessible from our user account. To establish this, we'll opt for another method to interact with the cluster by opening an SSH shell or terminal.

To open an SSH shell, navigate to "Interactive Apps" in the cluster's web front end's top menu. Then, select "HiPer4All Shell Access". A new browser tab will open with a fresh shell ready. It will appear as a black screen, which is normal.

Figure 2: The SSH shell opens in a new tab.

The SSH shell allows you to issue commands to the cluster's operating system. You can use it to install software, manage files, and execute scripts. In the upcoming step, we'll use the SSH shell to set up a Python virtual environment. For now, see if you can familiarize yourself with the shell by trying the following things:

- List the files in your home directory with `ls`.

- What is the difference between `ls` and `ls -a`?

- See if you can navigate within folders using cd `cd` command

- What does the command `pwd` do?

- What happens when you type `exit` into the SSH terminal?

- What does the command `man` do?

- Use the `man` command to look up the `cp` and `rm` commands!

Step 3: Create or copy an existing Python virtual environment

To work with Jupyter notebooks, we need a Python environment with the necessary packages installed. To create a Python environment, we have two options:

Option 1: Create a new Python virtual environment from scratch

Creating a fresh Python environment is a good idea if we want to start a new project from scratch. This way, we can ensure that only the necessary packages are installed. To create a new Python environment, follow these steps:

1. Navigate to your home directory on the cluster by typing `cd /cluster/user/<your_username>`. Replace `<your_username>` with your actual username.

2. Now, type `python3 -m venv env` to create a new Python virtual environment named `env`. This may take a few seconds to complete. A copy of a Python installation is created within the subfolder `env`.

3. Activate the Python virtual environment by typing `source env/bin/activate`. You should see the name of the virtual environment in the terminal prompt. This indicates that the virtual environment is active.

4. Install the necessary package for Juypter notebooks by typing `pip install jupyter`. This will install the package within your newly created virtual environment.

You can now move on to the next step.

Option 2: Copy an existing Python virtual environment

If you already have a Python virtual environment set up for a previous project, you can copy it to reuse it. This is useful if somebody took the time to install a required packages for a project. To copy an existing Python virtual environment, follow these steps:

1. Navigate to your home directory on the cluster by typing `cd /cluster/user/<your_username>`. Replace `<your_username>` with your actual username.

2. Copy the existing Python virtual environment by typing `cp -r /cluster/groups/<group_name>/env /cluster/user/<your_username>/env`. Replace `<group_name>` with the the group name for your course, which will be provided to you by your instructor. Also, replace `<your_username>` with your actual username. Copying the virtual environment will take a few minutes, depending on its size.

> Note
>
> The group name for the module "Big Data Analytics" in the summer term 24 is `l_aul_meseth_bda_ss24`.

3. Activate the Python virtual environment by typing `source env/bin/activate`. You should see the name of the virtual environment in the terminal prompt. This indicates that the virtual environment is active.

Since your prepared Python environment already has all necessary packages installed, you can proceed to the next step.

Step 4: Start a new Jupyter Lab server

Now that we have a Python environment set up, we can start a Jupyter Lab server to work with Jupyter notebooks. To start a Jupyter Lab server, follow these steps:

1. Go to the HPC cluster's web frontend and click on the "Jupyter Lab" app icon. It should be on the home screen. If not, you can find it under "Interactive Apps".

2. On the following screen, you need to input a few pieces of information for the server to run successfully:

   - **Account**: Select the group name for your course. This will be provided to you by your instructor.
   - **Python Virtual Environment**: Enter the path to your Python virtual environment from the previous step. This should be /cluster/user/<your_username>/env. Replace <your_username> with your actual username.
   - **Extra Modules**: In this input field, type "FFmpeg git", without the quotation marks.
   - **Max Time (in h)**: Set this to 4 hours. After this time, the server will shut down to free the resources for others to use.
   - **CPU Cores**: Set this to 4.
   - **GPU Type**: You can use 20 GB for most applications. However, when working with Large Language Models (LLMs), you may need more. In this case, you can select 80 GB.

3. Click on the "Launch" button to start the Jupyter Lab server. This may take a few minutes to set up. Once the server is running, the status in the upper right corner will change to "Running".

4. Finally, click on the "Connect to Jupyter Lab" button to open the Jupyter Lab interface in a new browser tab.

You successfully started a Jupyter Lab server when you see the Jupyter Lab interface as shown in the image below. of course, the files you see on the left panel may differ from the image.
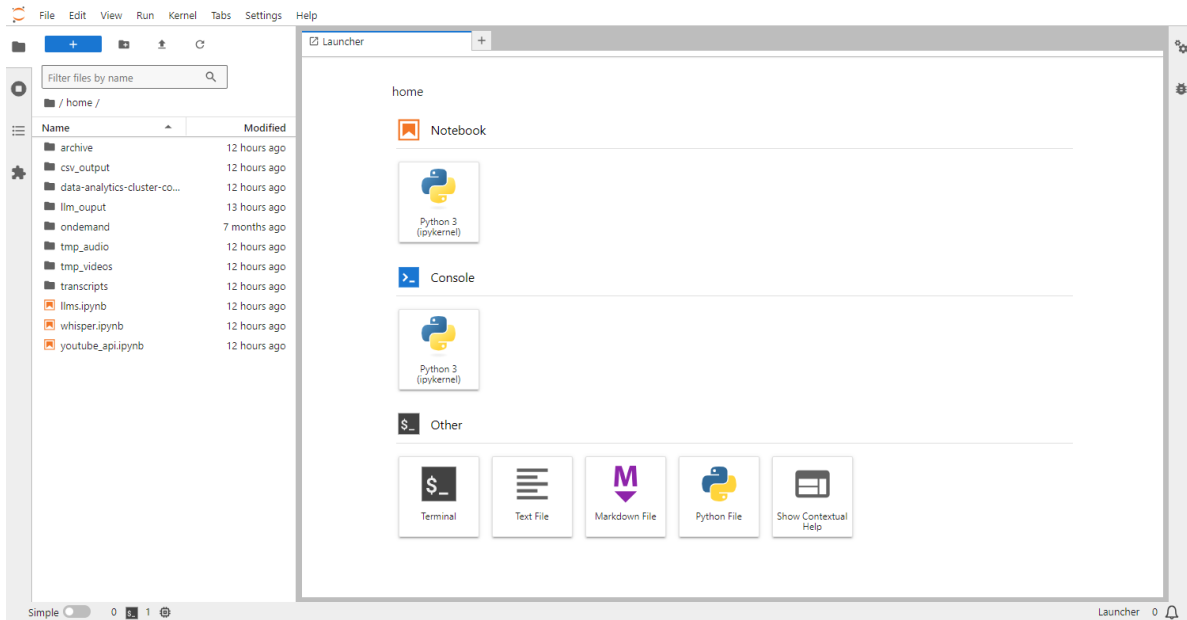
Figure 3: The Jupyter Lab interface

Step 5: Clone the code repository with example notebooks

To get you started with working on the HPC cluster, I prepared a code repository with several example notebooks. You can clone this repository to your Jupyter Lab environment by following these steps:

1. Return to the SSH shell tab in your browser.

2. Navigate to your home directory by typing `cd /cluster/user/<your_username>`. Replace `<your_username>` with your actual username.

3. Clone the code repository by typing:
   `git clone https://github.com/winf-hsos/data-analytics-cluster-code.git`
   This will copy the repository to a subfolder `/data-analytics-cluster-code` your current directory.

4. Go back to the Juypter Lab tab. You should now see the new repository when you navigate to the `/cluster` folder.

Step 6: Open and run the Python introduction notebook

In the top level folder of the code repository, you find a file called `python_introduction.ipynb`. This notebook provides an introduction to Python and Jupyter notebooks. Open this notebook

by double-clicking on it.

To complete this exercise, work your way through the notebook to get an quick overview of what you can do with Python and Jupyter notebooks. Write down any questions you have or things you'd like to learn more about. We'll discuss them in class.