

User Guide

1. Login

1. Open your browser and go to: <http://dgx.kmitonline.in:8000/hub/login>
2. Log in using your assigned **username** and **password**.
3. After successful login, you will be redirected to the **JupyterLab workspace**.

2. Default Server (CPU Only)

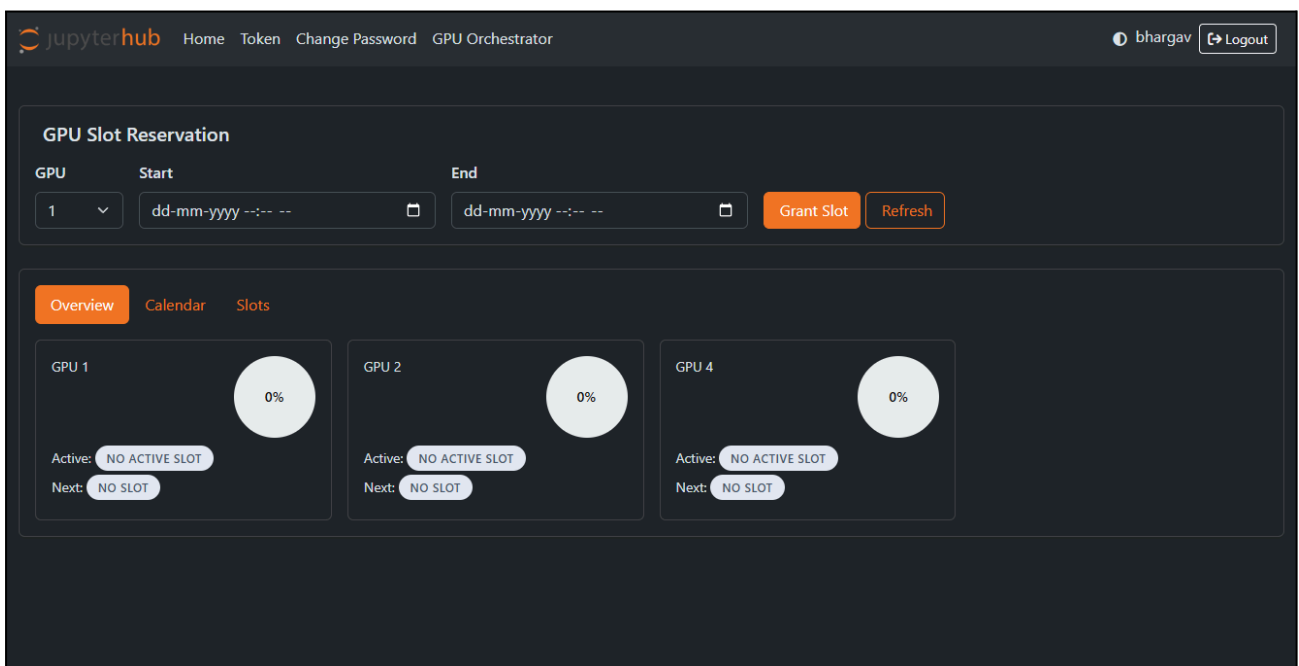
- When you log in for the first time (or after stopping your server), JupyterHub starts a **CPU-only server** by default.
- You can:
 - Write and run notebooks
 - Execute scripts
 - Perform general development tasks
- **No GPU is available** unless explicitly reserved.

3. When You Need a GPU

For GPU workloads (Deep Learning, large models, training, etc.), you must:

1. **Reserve a GPU** using the GPU Orchestrator page.
2. **Restart your server** so it starts on the reserved GPU.

4. GPU Orchestrator Page



The screenshot displays the JupyterHub GPU Orchestrator interface. At the top, there is a navigation bar with the JupyterHub logo, links for Home, Token, Change Password, and GPU Orchestrator, and a user profile for 'bhargav' with a Logout button. The main content area is titled 'GPU Slot Reservation' and features a form with a GPU dropdown menu set to '1', start and end date pickers, and 'Grant Slot' and 'Refresh' buttons. Below the form, there are three tabs: 'Overview', 'Calendar', and 'Slots'. The 'Overview' tab is active, showing three GPU status cards for GPU 1, GPU 2, and GPU 4. Each card displays a circular progress indicator at 0%, an 'Active' status of 'NO ACTIVE SLOT', and a 'Next' status of 'NO SLOT'.

Controls at the Top

- **GPU** – Dropdown (1, 2, or 4) to select the GPU

- **Start** – Start date & time (dd-mm-yyyy hh:mm)
- **End** – End date & time (dd-mm-yyyy hh:mm)
- **Grant Slot** – Create a reservation
- **Refresh** – Reload the current status

Tabs

1. **Overview**
 - Shows each GPU (1, 2, 4)
 - Displays:
 - **Active** slot (currently running)
 - **Next** slot (upcoming)
 - Circular graph showing percentage usage
2. **Calendar**
 - Day / Week / Month view
 - Shows all reservations for each GPU
3. **Slots**
 - Table view of all slots (past, current, and future)

5. How to Reserve a GPU

Step 5.1 – Open the GPU Orchestrator

- Open this URL after login: <http://dgx.kmitonline.in:8000/hub/gpu-orchestrator>

OR

- Either click the **File** at top bar/menu
- Locate and click on the **Hub Control Panel**. Open the **GPU Orchestrator**

Step 5.2 – Reserve a GPU

On the GPU Orchestrator page:

- Select:
 - **GPU**: choose one of the available GPUs from the list (e.g., GPU 1 / 2 / 4).
 - **Start time** and **End time**: the time period during which you want to use the GPU.
- Click **Grant Slot** for reservation.
- You should now see your slot listed under “**Active Slots / Upcoming Slots**”.

Note: Please reserve only the time you actually need so others can use the GPUs.

Step 5.3 – Start/Restart Your Server on GPU

Once your slot start time has begin:

1. Go back to the **Hub control panel**:

- Click on your username (top-right) → “**Home**”, or
 - Use the URL: <http://dgx.kmitonline.in:8000/hub/home>
2. If a server is already running:
 - Click “**Stop My Server**” and wait for it to fully stop.
 3. Then click “**Start My Server**” again.
 4. The GPU Orchestrator + TimeSlot system will:
 - Check your active reservations.
 - Start your user server on the **GPU you reserved**.

6. Verifying GPU Inside JupyterLab

1. Open a **Terminal** in JupyterLab. Execute the following command:
`nvdiia-smi`
2. You should see active slot **booked GPUs listed** (the one assigned to your slot).

7. Project Setup Instructions on DGX

Follow these steps to set up your project environment and organize your files inside the DGX server.

Step 7.1 – Open a Terminal in JupyterLab

1. In the JupyterLab interface, click on “**+ Launcher**” (plus icon) or **File** → **New Launcher**.
2. Under **Other**, click “**Terminal**” to open a new terminal session.

You will now be in your home directory inside the DGX container.

Step 7.2 – Create and Activate Your Conda Environment

You can either create a **new environment** or create one from an **existing env.yaml file**.

Option A – Create a New Environment (example: macrocyclesenv)

```
conda create -n macrocyclesenv python=3.10
```

Option B – Create an Environment from an Existing File

Make sure your **env.yaml** file is in the current directory, then run:

```
conda env create -f env.yaml
```

Activate the Environment

```
conda activate macrocyclesenv
```

Step 7.3 – Install Required Packages (Example: PyTorch)

With the environment activate, install PyTorch and related packages (example config):

```
conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c  
pytorch -c nvidia
```

Use the versions recommended by your course/project. Installation may take a few minutes.

8. Possible Issues & solutions

Issue: Failed to initialize NVML

Solution:

- Go to File → Hub Control Panel → Home
- Stop the server
- Start the server again

Issue: No devices were found

Solution:

- Ensure you have reserved a GPU slot:
 - File → Hub Control Panel → GPU Orchestrator
- Restart your server after the slot start time

Issue: GPU not visible even after reservation

Solution:

- Confirm:
 - Your slot is active
 - You restarted the server after slot start time
- Run `nvidia-smi` again
- If the issue persists, contact the admin

Note: *If you face any issue not listed in this document, or if the provided solutions do not resolve your problem, please contact the DGX admin immediately.*